《计算机组成原理》习题解答

第 1 章

1. 解释概念或术语：实际机器、虚拟机器，机器指令、机器指令格式，主机、CPU、主存、I/O、PC、IR、ALU、CU、AC、MAR、MDR，机器字长、存储字长、指令字长、CPI、T_C、主频、响应时间、吞吐量、MIPS、MFLOPS。

答：略

2. 如何理解计算机系统的层次结构？说明高级语言、汇编语言及机器语言的差别与联系。

答：⑴计算机系统是由软件和硬件结合而成的整体。为了提高计算机系统的好用性，程序设计语言的描述问题能力越来越强，各种程序设计语言大体上是一种层次结构，即高等级编程语言指令包含低等级编程语言指令的全部功能。

对于使用不同层次编程语言的程序员来说，他们所看到的同一计算机系统的属性是不同的，这些属性反映了同一计算机系统的不同层次的特征，即同一计算机系统可划分成多个层次结构，不同层次的结构反映的计算机系统的特征不同而已。

⑵机器语言是能够被计算机硬件直接识别和执行的程序设计语言，机器语言是一种面向硬件的、数字式程序设计语言；汇编语言和高级语言均用符号表示机器语言指令，指令很容易阅读和编写、但不能被硬件直接识别和执行，它们均是一种面向软件的、符号式程序设计语言；相对于汇编语言而言，高级语言描述问题的能力更强；高级语言和汇编语言程序必须翻译成机器语言程序后，才能在计算机硬件上执行。

3. 计算机系统结构、计算机组成的定义各是什么？两者之间有何关系？

答：计算机系统结构是指机器语言程序员或编译程序编写者所看到的计算机系统的属性，包括概念性结构和功能特性两个方面。主要研究计算机系统软硬件交界面的定义及其上下的功能分配。

计算机组成是指计算机硬件设计人员所看到的计算机系统的属性。主要研究如何合理地逻辑实现硬件的功能。

计算机组成是计算机系统结构的逻辑实现。

4. 冯·诺依曼模型的存储程序原理包含哪些内容、对计算机硬件和软件有哪些要求？冯·诺依曼模型计算机的特点有哪些？

答：存储程序原理是指程序和数据预先存放在存储器中，机器工作时自动按程序的逻辑顺序从存储器中逐条取出指令并执行。

存储程序原理要求存储器是由定长单元组成的、按地址访问的、一维线性空间结构的存储部件；要求软件指令支持用地址码表示操作数在存储器中的地址，指令长度为存储单元长度的倍数，编程语言中必须有转移型指令，以实现程序存储顺序到程序逻辑顺序的转变。

冯·诺依曼模型计算机的特点可归纳为如下几点：

⑴计算机由运算器、控制器、存储器、输入设备和输出设备组成；

⑵存储器是由定长单元组成的、按地址访问的、一维线性空间结构；

⑶程序由指令组成，指令和数据以等同地位存放在存储器中；

⑷机器工作时自动按程序的逻辑顺序从存储器中逐条取出指令并执行；

⑸指令由操作码和地址码组成，操作码用于表示操作的性质，地址码用于表示操作数在

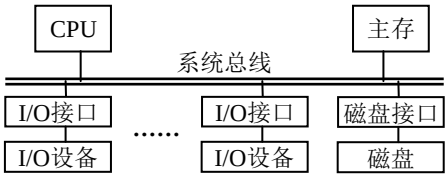
存储器中的地址；

⑹指令和数据均采用二进制方式表示，运算亦采用二进制方式；

⑺机器以运算器为中心，输入/输出设备与存储器间的数据传送都经过运算器。

5. 现代计算机均采用冯·诺依曼模型、但进行了改进，画出现代计算机硬件组成及结构图，并说明各部件的作用。

答：现代计算机结构大多在冯·诺依曼模型基础上进行了改进，以进一步提高系统的性能。改进主要包括以存储器为中心、多种存储器共存、采用总线互连三个方面。基本的硬件组成及结构图如下：



CPU 由运算器和控制器组成，运算器负责实现数据加工，实现算术逻辑运算；控制器负责指挥和控制各部件协调地工作，实现程序执行过程。

存储器由主存和辅存(如磁盘)组成，负责实现信息存储。主存由小容量、快速元器件组成，存放近期常用程序和数据；辅存由大容量、低价格元器件组成，存放所有的程序和数据；主存可被 CPU 直接访问，这样在提高访存速度的同时，可降低存储器总成本。

I/O 设备负责实现信息的输入和输出，以及信息的格式变换。

通过总线实现部件互连的好处是可以实现 CPU 的操作标准化，而操作标准化的具体实现部件是 I/O 接口，它负责缓冲和中转相关操作。

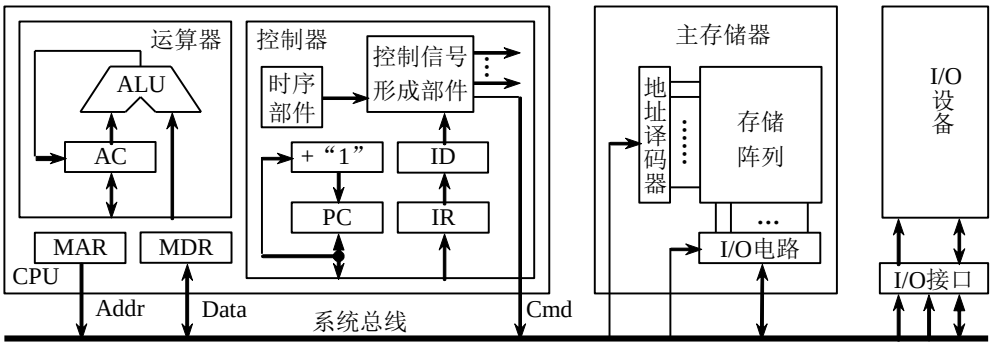
6. 若某计算机的机器指令格式如表 1.2 所示，请写出求 s=a+b+c 的机器语言程序，其中 a、b、c 存放在起始地址为 0000100000 的连续 3 个主存单元中，而 s 则要求存放到地址为 0000001000 的主存单元中。

解：假设程序第一条指令存放在第 1000000000 号存储单元中，则程序清单如下：

主存单元地址（二进制）	指令（二进制）		注　　释
	操 作 码	地 址 码	
…	…		
0000001000	*s*		结果数据 *s*
…	…		
0000100000	*a*		原始数据 *a*
0000100001	*b*		原始数据 *b*
0000100010	*c*		原始数据 *c*
…	…		
1000000000	000001	0000100000	取数 *a* 到累加器 AC 中
1000000001	000011	0000100001	(AC)＋*b*，结果存于 AC 中
1000000010	000011	0000100010	(AC)＋*c*，结果存于 AC 中
1000000011	000010	0000001000	将 AC 中内容存到 *s* 所在主存单元中
1000000100	000100	**********	停机，地址码空闲（值可任意）

7. 画出基于累加器 CPU 的主机框图，说明题 6 的机器语言程序的执行过程(尽可能详细)。简述执行过程与冯·诺依曼模型的存储程序原理的关系。

答：基于累加器 CPU 的的主机框图如下：



假设 s=a+b+c 程序已被调入主存、首指令地址已写入到 PC 中，即(PC)=1000000000。程序运行启动后，计算机硬件自动地、逐条地、按(PC)为指令地址实现取指令、分析指令、执行指令的对应操作，直到执行到停机指令为止。假设 IR 中操作码记为 OP(IR)、地址码记为 AD(IR)，则 s=a+b+c 程序执行过程的具体操作如下：

```
（1）PC→MAR、MAR→ABus、Read→CBus          ；MAR＝PC＝1000000000，   取指令开始
（2）WMFC，(PC)＋1→PC                       ；PC＝1000000001（下条指令地址）
（3）MDR→IR                                 ；IR＝000001 0000100000，取指令完成
    ID 对 OP(IR)译码                        ；CU 得知当前为取数指令
（4）AD(IR)→MAR、MAR→ABus、Read→Cbus        ；MAR＝0000100000，执行指令开始
（5）WMFC
（6）MDR→AC                                 ；AC＝MDR＝a，      执行指令完成
（7）PC→MAR、MAR→ABus、Read→CBus            ；MAR＝PC＝1000000001，  取指令开始
（8）WMFC，(PC)＋1→PC                       ；PC＝1000000010（下条指令地址）
（9）MDR→IR                                 ；IR＝000011 0000100001，取指令完成
    ID 对 OP(IR)译码                        ；CU 得知当前为加法指令
（10）AD(IR)→MAR、MAR→ABus、Read→CBus       ；MAR＝0000100001，执行指令开始
（11）WMFC
（12）(MDR)＋(AC)→AC                        ；AC＝a＋b，        执行指令完成
（13）PC→MAR、MAR→ABus、Read→CBus           ；MAR＝PC＝1000000010，取指令开始
（14）WMFC，(PC)＋1→PC                      ；PC＝1000000011（下条指令地址）
（15）MDR→IR                                ；IR＝000011 0000100010，取指令完成
    ID 对 OP(IR)译码                        ；CU 得知当前为加法指令
（16）AD(IR)→MAR、MAR→ABus、Read→CBus       ；MAR＝0000100010，执行指令开始
（17）WMFC
（18）(MDR)＋(AC)→AC                        ；AC＝a＋b+c，      执行指令完成
（19）PC→MAR、MAR→ABus、Read→CBus           ；MAR＝PC＝1000000011，  取指令开始
（20）WMFC，(PC)＋1→PC                      ；PC＝1000000100（下条指令地址）
（21）MDR→IR                                ；IR＝000010 0000001000，取指令完成
    ID 对 OP(IR)译码                        ；CU 得知当前为存数指令
（22）AD(IR)→MAR、MAR→ABus、Write→Cbus      ；MAR＝0000100000，执行指令开始
（23）AC→MDR、MDR→DBus、WMFC                ；MDR＝AC＝a＋b+c，执行指令完成
（24）PC→MAR、MAR→ABus、Read→Cbus           ；MAR＝PC＝1000000100，   取指令开始
（25）WMFC，(PC)＋1→PC                      ；PC＝1000000101（下条指令地址）
（26）MDR→IR                                ；IR＝000100 **********，取指令完成
    ID 对 OP(IR)译码                        ；CU 得知当前为停机指令
（27）机器自动停机                          ；执行停机指令完成
```

从程序执行过程可以看出：由于指令存放在存储器中，故指令执行过程分为取指令(含

分析指令)、执行指令两个阶段；由于存储器同时只接收一个访问操作，故程序执行过程是循环的指令执行过程，循环变量为 PC 中的指令地址；只要按照程序逻辑顺序改变(PC)，可以实现按程序逻辑顺序执行程序的目标。

8. 指令和数据均存放在存储器中，计算机如何区分它们？

答：由于存储器访问只使用地址和命令(Read/Write)信号，而指令和数据均以二进制编码形成存放在存储器中，因此，从存储器取得的信息本身是无法区分是指令还是数据的。

计算机只能通过信息的用途来区分，即取指令时取得的是指令，指令执行时取操作数或写结果对应的信息是数据。即计算机通过程序执行过程或指令执行过程的不同阶段来区分。

9. 在某 CPU 主频为 400MHz 的计算机上执行程序 A，程序 A 中指令类型、执行数量及平均时钟周期数如下表所示。

指令类型	指令执行数量	平均时钟周期数(/指令)
整数	45000	1
数据传送	75000	2
浮点数	8000	4
条件转移	1500	2

求该计算机执行程序 A 时的程序执行时间、平均 CPI 及 MIPS。

解：CPU 时钟周期 $T_C=1/f=1/(400\times10^6)=2.5\text{ns}$

程序执行时间 $T_{CPU}=[45000\times1+75000\times2+8000\times4+1500\times2]\times2.5=0.575\text{ms}$。

平均 CPI＝(45000×1＋75000×2＋8000×4＋1500×2)

÷(45000＋75000＋8000＋1500)

＝1.776(时钟周期/指令)

MIPS＝(45000＋75000＋8000＋1500)/ $(0.575\times10^{-3}\times10^6)$＝225.2 百万条/秒

10. 冯·诺依曼模型计算机的性能瓶颈有哪些？简述缓解性能瓶颈严重性的方法。

答：冯·诺依曼模型计算机的性能瓶颈有 CPU-MEM 瓶颈、指令串行执行瓶颈两个。

对缓解 CPU-MEM 瓶颈而言，主要目标是减少 MEM 访问延迟、提高 MEM 传输带宽，常用的方法有采用多种存储器构成层次结构存储系统、采用多级总线互连、采用并行结构存储器等。

对缓解指令串行执行瓶颈而言，主要目标是尽可能实现并行处理，常用的方法有采用流水线技术、数据流技术、超标量技术、超线程技术、多核技术等。

第 2 章

1. 解释概念或术语：进制、机器数、原码、补码、移码、变形补码、BCD 码、交换码、内码、奇校验、CRC、上溢、下溢、左规、对阶、溢出标志、进位标志、部分积、Booth 算法、交替加减法除法、警戒位、全加器、并行加法器、行波进位、先行进位。

答：略

2. 完成下列不同进制数之间的转换

（1）$(347.625)_{10}=(\quad)_2=(\quad)_8=(\quad)_{16}$

（2）$(9C.E)_{16}=(\quad)_2=(\quad)_8=(\quad)_{10}$

（3）$(11010011)_2=(\quad)_{10}=(\quad)_{8421BCD}$

解：（1）$(347.625)_{10}=(101011011.101)_2=(533.5)_8=(15B.A)_{16}$

（2）$(9C.E)_{16}=(10011100.1110)_2=(234.7)_8=(156.875)_{10}$

（3）$(11010011)_2=(211)_{10}=(001000010001)_{8421BCD}$

3. 对下列十进制数，分别写出机器数长度为 8 位（含 1 位符号位）时的原码及补码。

（1）＋23/128　（2）－35/64　（3）43　（4）－72

（5）＋7/32　（6）－9/16　（7）＋91　（8）－33

解：（1）$[+23/128]_{原}=0.0010111$，$[+23/128]_{补}=0.0010111$；

（2）$[-35/64]_{原}=1.1000110$，$[-35/64]_{补}=1.0111010$；

（3）$[43]_{原}=00101011$，$[43]_{补}=00101011$；

（4）$[-72]_{原}=11001000$，$[-72]_{补}=10111000$；

（5）$[+7/32]_{原}=0.0011100$，$[+7/32]_{补}=0.0011100$；

（6）$[-9/16]_{原}=1.1001000$，$[-9/16]_{补}=1.0111000$；

（7）$[+91]_{原}=01011011$，$[+91]_{补}=01011011$；

（8）$[-33]_{原}=10100001$，$[-33]_{补}=11011111$。

4. 对下列机器数（含 1 位符号位），若为原码时求补码及真值，若为补码或反码时求原码及真值。

（1）$[X]_{原}=100011$　（2）$[X]_{补}=0.00011$　（3）$[X]_{反}=1.01010$

（4）$[X]_{原}=1.10011$　（5）$[X]_{补}=101001$　（6）$[X]_{反}=101011$

解：（1）$[X]_{补}=111101$，$X=-00011=-3$；

（2）$[X]_{原}=0.00011$，$X=+0.00011=+3/32$；

（3）$[X]_{原}=1.10101$，$X=-0.10101=-21/32$；

（4）$[X]_{补}=1.01101$，$X=-0.10011=-19/32$；

（5）$[X]_{原}=110111$，$X=-10111=-23/32$；

（6）$[X]_{原}=110100$，$X=-10100=-20/32$。

5. （1）若$[X]_{补}=1.01001$，求$[-X]_{补}$及 X；

（2）若$[-X]_{补}=101001$，求$[X]_{补}$及 X。

解：（1）$[-X]_{补}=0.10111$，$X=-0.10111=-23/32$；

（2）$[X]_{补}=010111$，$X=+10111=+23$。

6. （1）若 $X=+23$ 及 -42，分别求 8 位长度的$[X]_{移}$；

（2）若$[X]_{移}$=1100101 及 0011101，分别求 X。

解：（1）$[+23]_{移}$=10010111，$[-42]_{移}$=01010110；

（2）$[X]_{移}$=1100101 时的 X=+100101=+37，

$[X]_{移}$=0011101 时的 X=-100011=-35。

7. 若$[X]_{补}=0.x_{-1}x_{-2}x_{-3}x_{-4}x_{-5}$，$[Y]_{补}=1y_4y_3y_2y_1y_0$，求下列几种情况时，$x_{-i}$或 y_i的取值。

（1）$X>1/4$　（2）$1/8\geqslant X>1/16$　（3）$Y<-16$　（4）$-32<Y\leqslant-8$

解：（1）$[1/4]_{补}=0.01000$，

故$[(x_{-1}=0)\wedge(x_{-3}=1\vee x_{-4}=1\vee x_{-5}=1)]\vee(x_{-1}=1)$时 $X>1/4$；

（2）$[1/8]_{补}=0.00100$，$[1/16]_{补}=0.00010$，

故$(x_{-1}=0\wedge x_{-2}=0)\wedge[(x_{-3}=1\wedge x_{-4}=0\wedge x_{-5}=0)\vee(x_{-3}=0\wedge x_{-4}=1\wedge x_{-5}=1)]$时 $1/8\geqslant X>1/16$；

（3）$[-16]_{补}$=110000，故 $y_4=0$ 时 $Y<-16$；

（4）$[-8]_{补}$=111000，$[-32]_{补}$=100000，故$(y_4=1\wedge y_3=1\wedge y_2=0\wedge y_1=0\wedge y_0=0)\vee(y_4\oplus y_3=1)\vee[y_4=0\wedge y_3=0\wedge(y_2=1\vee y_1=1\vee y_0=1)]$时 $-32<Y\leqslant-8$。

8. 冗余校验的基本原理是什么？

答：数据发送时，除发送数据信息外，还冗余发送按某种规律形成的校验信息；数据接收时，用所接收数据信息形成新的校验信息，与所接收的校验信息比较，以此判断是否发生了错误，出错时报告出错或自动校正错误。

9. 若采用奇校验，下述两个数据的校验位的值是多少？

（1）0101001　（2）0011011

答：（1）数据 0101001 的奇校验位值为 0⊕1⊕0⊕1⊕0⊕0⊕1⊕1=0；

（2）数据 0011011 的奇校验位值为 0⊕0⊕1⊕1⊕0⊕1⊕1⊕1=1。

10. 若下列奇偶校验码中只有一个有错误，请问采用的是奇/偶校验？为什么？

（1）10001101　（2）01101101　（3）10101001

答：上述奇偶校验码采用的是偶校验编码方式。

由于三个奇偶校验码中分别有偶数、奇数、偶数个“1”，而只有一个校验码有错误，

故第 2 个奇偶校验码(01101101)有错误；

又由于第 2 个奇偶校验码有奇数个“1”，故校验码采用的是偶校验编码方式。

11. 设有 8 位数据信息 01101101，请写出求其海明校验码的过程。

解：本题中数据位数 n=8，数据信息 $m_8\cdots m_1$=01101101，设检验信息位数为 k 位，

（1）先求得校验信息位数 k，根据 $2^k-1\geqslant 8+k$ 的要求，可得 k=4 位；

（2）列出 $n+k$=8+4=12 位校验码中的信息排列：$m_8\ m_7m_6m_5p_4m_4m_3m_2p_3m_1p_2p_1$。

（3）设各校验组采用偶校验编码方式，各校验组校验位的值为：

$p_4=m_8\oplus m_7\oplus m_6\oplus m_5=0\oplus1\oplus1\oplus0=0$，

$p_3=m_8\oplus m_4\oplus m_3\oplus m_2=0\oplus1\oplus1\oplus0=0$，

$p_2=m_7\oplus m_6\oplus m_4\oplus m_3\oplus m_1=1\oplus1\oplus1\oplus1\oplus1=1$，

$p_1=m_7\oplus m_5\oplus m_4\oplus m_2\oplus m_1=1\oplus0\oplus1\oplus0\oplus1=1$；

（4）海明偶校验码为：011001100111。

12. 若机器数表示时字长为 8 位，写出下列情况时它能够表示的数的范围（十进制）。

（1）无符号整数；　（2）原码编码的定点整数；

（3）补码编码的定点整数；　（4）原码编码的定点小数；

（5）补码编码的定点小数。

解：（1）无符号整数的表示范围是 00000000～11111111，即 0～255；

（2）原码定点整数的表示范围是-1111111～+1111111，即-127～+127；

（3）补码定点整数的表示范围是-(1111111+1)～+1111111，即-128～+127；

（4)原码定点小数的表示范围是-0.1111111～+0.1111111,即-127/128～+127/128；

（5)补码定点小数的表示范围是-1.0000000～+0.1111111,即-128/128～+127/128。

13. 对两个 8 位字长的定点数 9BH 及 FFH，分别写出它们采用原码编码、补码编码及移码编码时的十进制整数的真值，并写出它们表示为无符号数时的十进制真值。

解：机器码	9BH	FFH
原码编码的真值(整数)	-27	-127
补码编码的真值(整数)	-101	-1
移码编码的真值(整数)	+27	+127
无符号编码的真值(整数)	155	255

14. 若浮点数表示格式（从高位到低位）为：阶码 6 位（含 1 位阶符）、尾数 10 位（含 1 位数符），请写出 51/128、－27/1024、7.375、－86.5 所对应的机器数。

（1）阶码和尾数均为原码；

（2）阶码和尾数均为补码；

（3）阶码为移码、尾数为补码。

解：（1）阶码和尾数均为原码时，

$[51/128]_{浮}$=[0.0110011]$_{浮}$=100001 0110011000 或 000000 0011001100 或…，

$[-27/1024]_{浮}$=[-0.0000011011]$_{浮}$=100101 1110110000 或 100001 1000011011 或…，

$[7.375]_{浮}$=[111.011]$_{浮}$=000011 0111011000 或 000110 0000111011 或…，

$[-86.5]_{浮}$=[-1010110.1]$_{浮}$=000111 1101011010 或 001000 1010101101 或…；

（2）阶码和尾数均为补码时，

$[51/128]_{浮}$=111111 0110011000 或 000000 0011001100 或…，

$[-27/1024]_{浮}$=111011 1001010000 或 111111 1111100101 或…，

$[7.375]_{浮}$=000011 0111011000 或 000110 0000111011 或…，

$[-86.5]_{浮}$=000111 1010100110 或 001000 1101010011 或…；

（3）阶码为移码、尾数为补码时，

$[51/128]_{浮}$=011111 0110011000 或 100000 0011001100 或…，

$[-27/1024]_{浮}$=011011 1001010000 或 011111 1111100101 或…，

$[7.375]_{浮}$=100011 0111011000 或 100110 0000111011 或…，

$[-86.5]_{浮}$=100111 1010100110 或 101000 1101010011 或…。

15. 若浮点数表示格式采用 6 位阶码（含 1 位阶符）、10 位尾数（含 1 位数符），阶码和尾数均采用补码编码。

（1）写出浮点数能表示的正数及负数的范围；

（2）写出规格化浮点数能表示的正数及负数的范围。

解：（1）浮点数正数区的范围为：$+2^{-9}\times2^{-32}$～$+(1-2^{-9})\times2^{+31}$，

浮点数负数区的范围为：$-1\times2^{+31}$～$-2^{-9}\times2^{-32}$；

（2）规格化浮点数正数区的范围为：$+2^{-1}\times2^{-32}$～$+(1-2^{-9})\times2^{+31}$，

规格化浮点数负数区的范围为：$-1\times2^{+31}$～$-(2^{-1}+2^{-9})\times2^{-32}$。

16. 若浮点数表示格式为：6 位阶码（含 1 位阶符）、10 位尾数（含 1 位数符）。分别写

出阶码和尾数均为原码及均为补码时，下列数值为规格化数时的机器码。

（1）＋51/128　（2）－51/128　（3）－1/64

解：（1）阶码和尾数均为原码时，规格化数的机器码为 100001 0110011000，

阶码和尾数均为补码时，规格化数的机器码为 111111 0110011000；

（2）阶码和尾数均为原码时，规格化数的机器码为 100001 1110011000，

阶码和尾数均为补码时，规格化数的机器码为 111111 1001101000；

（3）阶码和尾数均为原码时，规格化数的机器码为 100101 1100000000，

阶码和尾数均为补码时，规格化数的机器码为 111010 1000000000。

17. 若机器中单精度浮点数采用 IEEE 754 标准表示。

（1）对机器码为$(99D00000)_{16}$及$(59800000)_{16}$的浮点数，请写出它们的真值；

（2）请写出－51/128 的机器码。

解：（1）由于机器码$(99D00000)_{16}$＝1 00110011 10100000000000000000000B，

故浮点数的符号码 S＝1、阶码 E＝00110011、尾数码 M＝10100000000000000000000，

因 1＜E＜255，故机器码表示的为规格化浮点数，

$(99D00000)_{16}$的真值 $N=(-1)^1\times2^{51-127}\times1.10100000000000000000000=-0.1101\times2^{-76}$；

由于机器码$(59800000)_{16}$=0 10110011 00000000000000000000000B，

故浮点数的符号码 S＝0、阶码 E＝10110011、尾数码 M＝00000000000000000000000，

因 1＜E＜255，故机器码表示的为规格化浮点数，

$(59800000)_{16}$的真值 $N=(-1)^0\times2^{179-127}\times1.\ 00000000000000000000000=+0.1\times2^{+53}$。

（2）$(-51/128)_{10}=(-0.0110011)_2=(-1)^1\times(1.10011)\times2^{125-127}$，

则用 IEEE754 标准表示时，符号码 S＝1、阶码 E＝125、尾数 M＝0.10011，

故－51/128 的单精度浮点数机器码为 1 01111101 10011000000000000000000。

18. 字符在计算机中的表示可看作无符号定点整数，对字符的操作有比较是否相同、判断前后次序等关系运算，需要哪些支持才能用算术运算和逻辑运算实现关系运算？

答：由于字符数据可看作无符号定点整数，故字符操作的结果可以用两个无符号定点整数关系运算的结果表示。

设 NA 及 NB 为无符号定点整数，NC 为有符号定点整数，且 NA－NB＝NC，

则①当 NA＞NB 时，NC 的符号为正，

②当 NA＜NB 时，NC 的符号为负，

③当 NA＝NB 时，NC 的值为零，

④当 NA≥NB 时，NC 的符号为正、或者 NC 的值为零，

⑤当 NA≤NB 时，NC 的符号为负、或者 NC 的值为零；

即对算术运算(减法)结果的符号、是否为零进行逻辑运算(逻辑与、逻辑或)，就可以得到关系运算的结果。

因此，运算器中设置“结果符号是否为负”及“结果是否为零”两个标志位，并且有对这 2 个硬件标志位的 5 种逻辑操作硬件时，就可以用算术运算和逻辑运算实现关系运算了。

19. 各种应用数据在计算机中一般表示成哪几种数据类型？对某个机器数，如何才能够知道它的数据类型？

答：计算机中的应用数据一般有数值数据和非数据数据两大类型，数值数据的运算均为算术运算，数据可表示为定点数或浮点数两种数据类型；非数值数据的运算比较复杂，可能为逻辑运算，或算术运算或关系运算，数据可表示为逻辑数，或定点数或浮点数。故应用数据在计算机中一般表示成定点数、浮点数及逻辑数三种数据类型。

由于计算机中均用二进制表示数据和指令，只能通过约定方式隐含表示符号及小数点

等。而这种约定只在数据操作时才有实际意义，因此，对于某机器数，从数据本身无法知道它的数据类型，只能通过对其操作的指令来表明这个数的数据类型。如对 32 位机器数 99D00000H，当它为浮点运算指令的操作数时，它是浮点数；当它为定点运算指令的操作数时，它是定点数。

20. 若 8 位机器码为 0010100，请问逻辑左移多少次后溢出？逻辑右移多少次后再左移同样多次数时机器码开始不同？请分别说明原因。

答：逻辑左移 3 位后溢出，因为左起第一个“1”被移丢，故溢出。

逻辑右移 3 位后再逻辑左移 3 位时机器码与原来不同，因为机器码 0010100→0000010→0010000，右起第一个“1”被移丢，损失精度后再左移 3 位机器码发生变化。

21. 设机器数字长为 8 位(含 1 位符号)，分别写出对下列机器数算术左移 1 位、2 位，算术右移 1 位、2 位的结果，并说明结果是否正确。

$[X]_{原}=0.0011010$；$[X]_{补}=1.1101000$；$[X]_{反}=1.0101111$；

$[X]_{原}=1.0011010$；$[X]_{补}=1.1001101$；$[X]_{反}=1.1001110$

解：结果见下表，其中√表示结果正确、×表示结果溢出、∠表示结果精度受损失。

	算术左移 1 位	算术左移 2 位	算术右移 1 位	算术右移 2 位
$[X]_{原}=0.0011010$	0.0110100 √	0.1101000 √	0.0001101 √	0.0000110 ∠
$[X]_{原}=1.0011010$	1.0110100 √	1.1101000 √	1.0001101 √	1.0000110 ∠
$[X]_{补}=1.1101000$	1.1010000 √	1.0100000 √	1.1110100 √	1.1111010 √
$[X]_{补}=1.1001101$	1.0011010 √	1.0110100 ×	1.1100110 ∠	1.1110011 ∠
$[X]_{反}=1.0101111$	1.1011111 ×	1.0111111 ×	1.1010111 √	1.1101011 √
$[X]_{反}=1.1001110$	1.0011101 √	1.0111011 ×	1.1100111 ∠	1.1110011 ∠

22. 若$[X]_{补}=x^S x_{n-1}\cdots x_0$，请推导$[2X]_{补}=2[X]_{补}$及$[\frac{1}{2}X]_{补}=x^S*2^{n-1}+\frac{1}{2}[X]_{补}$。

解：(1)因$[2X]_{补}=2^n+2X\equiv 2^n+2^n+2X=2\times(2^n+X)=2[X]_{补}$，

故$[2X]_{补}=2[X]_{补}$。

(2)当$X\geqslant 0$时，$x^S=0$，$[X]_{补}=2^n+X=0x_{n-1}\cdots x_0$，$X=+x_{n-1}\cdots x_0$，

则$[\frac{1}{2}X]_{补}=2^n+\frac{1}{2}X=2^n+x_{n-1}\cdots x_0/2=0x_{n-1}\cdots x_0/2=\frac{1}{2}[X]_{补}$；

$=0*2^{n-1}+\frac{1}{2}[X]_{补}=x^S*2^{n-1}+\frac{1}{2}[X]_{补}$；

当$X<0$时，$x^S=1$，$X=[X]_{补}-2^n=1x_{n-1}\cdots x_0-2^{n-1}-2^{n-1}=-2^{n-1}+x_{n-1}\cdots x_0$，

则$[\frac{1}{2}X]_{补}=2^n+(-2^{n-1}+x_{n-1}\cdots x_0)/2$

$=2^{n-1}+2^{n-1}+(-2^{n-1}+x_{n-1}\cdots x_0)/2=2^{n-1}+(2^{n-1}+x_{n-1}\cdots x_0)/2$

$=2^{n-1}+1x_{n-1}\cdots x_0/2=2^{n-1}+\frac{1}{2}[X]_{补}=x^S*2^{n-1}+\frac{1}{2}[X]_{补}$。

故对任意X，均有$[\frac{1}{2}X]_{补}=x^S*2^{n-1}+\frac{1}{2}[X]_{补}$。

23. 若机器数字长为 8 位(含 1 位符号)，请用补码运算规则计算下列各题。

（1）$A=9/64$，$B=-13/32$，求 $A+B$;

（2）$A=19/32$，$B=-18/128$，求 $A-B$;

（3）$A=-87$，$B=13$，求 $A-B$;

（4）$A=115$，$B=-24$，求 $A+B$

解：（1）因 $A=+0.0010010$、$B=-0.0110100$，则$[A]_{补}=0.0010010$、$[B]_{补}=1.1001100$，

$[A+B]_{补}=[A]_{补}+[B]_{补}=0.0010010+1.1001100=1.1011110$，

则 $A+B=-0.0100010=-17/64$;

（2）因 $A=+0.1001100$、$B=-0.0010010$，则$[A]_{补}=0.1001100$、$[-B]_{补}=0.0010010$，
$[A-B]_{补}=[A]_{补}+[-B]_{补}=0.1001100+0.0010010=0.1011110$，
则 $A-B=0.1011110=47/64$；

（3）因 $A=-1010111$、$B=+0001101$，则$[A]_{补}=1\ 0101001$、$[-B]_{补}=1\ 1110011$，
$[A-B]_{补}=[A]_{补}+[-B]_{补}=1\ 0101001+1\ 1110011=1\ 0011100$，
则 $A-B=-100$；

（4）因 $A=+1110011$、$B=-0011000$，则$[A]_{补}=0\ 1110011$，$[B]_{补}=1\ 1101000$，
$[A+B]_{补}=[A]_{补}+[B]_{补}=0\ 1110011+1\ 1101000=0\ 1011011$，
则 $A+B=+91$。

24. 若机器数字长为 6 位(含 1 位符号)，请用补码计算 $A+B$，并判断结果是否溢出。

（1）$A=0.11011$，$B=0.00011$；　（2）$A=0.11011$，$B=-0.10101$；

（3）$A=-0.10111$，$B=-0.01011$；　（4）$A=0.10011$，$B=0.01111$

解：（1）由题意$[A]_{补}=0.11011$，$[B]_{补}=0.00011$，
$[A+B]_{补}=[A]_{补}+[B]_{补}=0.11011+0.00011=0.11110$，$A+B=+0.11110$，
$[A+B]_{补}$的溢出标志 OVR=(0⊕0)(0⊕0)=0，故 $A+B$ 结果不溢出；

（2）由题意$[A]_{补}=0.11011$，$[B]_{补}=1.01011$，
$[A+B]_{补}=[A]_{补}+[B]_{补}=0.11011+1.01011=0.00110$，$A+B=+0.00110$，
$[A+B]_{补}$的溢出标志 OVR=(0⊕0)(1⊕0)=0，故 $A+B$ 结果不溢出；

（3）由题意$[A]_{补}=1.01001$，$[B]_{补}=1.10101$，
$[A+B]_{补}=[A]_{补}+[B]_{补}=1.01001+1.10101=0.11110$，$A+B=+0.11110$，
$[A+B]_{补}$的溢出标志 OVR=(1⊕0)(1⊕0)=1，故 $A+B$ 结果溢出；

（4）由题意$[A]_{补}=0.10011$，$[B]_{补}=0.01111$，
$[A+B]_{补}=[A]_{补}+[B]_{补}=0.10011+0.01111=1.00010$，$A+B=-0.11110$，
$[A+B]_{补}$的溢出标志 OVR=(0⊕1)(0⊕1)=1，故 $A+B$ 结果溢出。

25. 若机器数字长为 7 位(含 2 位符号)，请用变形补码计算 $A-B$，并判断结果是否溢出。

（1）$A=0.11011$，$B=-0.11111$；　（2）$A=0.10111$，$B=-0.01010$

解：（1）由题意$[A]_{变补}=00.11011$，$[B]_{变补}=11.00001$，$[-B]_{变补}=00.11111$，
则$[A-B]_{变补}=[A]_{变补}+[-B]_{变补}=00.11011+00.11111=01.11010$，
$A+B=-0.00110$，
$[A-B]_{变补}$的溢出标志 OVR=0⊕1=1，故 $A-B$ 结果溢出；

（2）由题意$[A]_{变补}=00.10111$，$[B]_{变补}=11.10110$，$[-B]_{变补}=00.01010$，
则$[A-B]_{变补}=[A]_{变补}+[-B]_{变补}=00.10111+00.01010=01.00001$，
$A+B=-0.00001$，
$[A-B]_{变补}$的溢出标志 OVR=0⊕1=1，故 $A-B$ 结果溢出。

26. 对下列 A 和 B，请用原码一位乘法求 $A\times B$。

（1）$A=0.110111$，$B=-0.101110$；　（2）$A=19$，$B=35$

解：（1）由题意$[A]_{原}=0.110111$，$[B]_{原}=1.101110$，$|A|=0.110111$，$|B|=0.101110$，
$[A\times B]_{原}$的符号位为 0⊕1=1，
按原码一位乘法规则，$|A|\times|B|$需进行 6 次判断-加法-移位操作，其过程如下表所示：

循环次数	部分积高位	乘数 （及部分积低位）	说明
6	0.000000	1 0 1 1 1 0	初始部分积 P_0=0.000000
	＋ 0.000000		乘数最低位为 0，应＋0
	0.000000		
5	0.000000	0 1 0 1 1 1	部分积及乘数同时右移 1 位
	＋ 0.110111		乘数最低位为 1，应＋|A|
	0.110111		
4	0.011011	1 0 1 0 1 1	部分积及乘数同时右移 1 位
	＋ 0.110111		乘数最低位为 1，应＋|A|
	1.010010		
3	0.101001	0 1 0 1 0 1	部分积及乘数同时右移 1 位
	＋ 0.110111		乘数最低位为 1，应＋|A|
	1.100000		
2	0.110000	0 0 1 0 1 0	部分积及乘数同时右移 1 位
	＋ 0.000000		乘数最低位为 0，应＋0
	0.110000		
1	0.011000	0 0 0 1 0 1	部分积及乘数同时右移 1 位
	＋ 0.110111		乘数最低位为 1，应＋|A|
	1.001111		
0	0.100111	1 0 0 0 1 0	部分积及乘数同时右移 1 位

即$|A|\times|B|$=0.100111100010，故$[A\times B]_{原}$=1.100111100010。

（2）由题意，$[A]_{原}$=0010011，$[B]_{原}$=0100011，$|A|$=010011，$|B|$=100011，

$[A\times B]_{原}$的符号位为 0⊕0=0，

$|A|\times|B|$需进行 6 次判断-加法-移位操作，其过程如下表所示：

循环次数	部分积高位	乘数 （及部分积低位）	说明
6	000000	1 0 0 0 1 1	初始部分积 P_0=000000
	＋ 010011		乘数最低位为 1，应＋|A|
	0 010011		6 位加法，0 为加法器的进位
5	001001	1 1 0 0 0 1	部分积及乘数同时右移 1 位
	＋ 010011		乘数最低位为 1，应＋|A|
	0 011100		
4	001110	0 1 1 0 0 0	部分积及乘数同时右移 1 位
	＋ 000000		乘数最低位为 0，应＋0
	0 001110		
3	000111	0 0 1 1 0 0	部分积及乘数同时右移 1 位
	＋ 000000		乘数最低位为 0，应＋0
	0 000111		
2	000011	1 0 0 1 1 0	部分积及乘数同时右移 1 位
	＋ 000000		乘数最低位为 0，应＋0
	0 000011		
1	000001	1 1 0 0 1 1	部分积及乘数同时右移 1 位
	＋ 010011		乘数最低位为 1，应＋|A|
	0 010100		
0	001010	0 1 1 0 0 1	部分积及乘数同时右移 1 位

即$|A|\times|B|$=001010011001，故$[A\times B]_{原}$=0 001010011001。

27. 若 A=0.011011，B=-0.100110，请用原码两位乘法求 $A\times B$。

解：由题意$[A]_{原}$=0.011011，$[B]_{原}$=1.100110，$|B|$=0.100110，

$[|A|]_{补}$=0.011011，$[-|A|]_{补}$=1.100101，$[2|A|]_{补}$=0.110110，

$[A\times B]_{原}$的的符号位为 0⊕1=1，

由于|B|为 6 位(偶数个)，乘法运算时需在|B|的最高位前增加两个 0，以处理乘法运算结束时可能的 T=1 的情况，故共循环 4 次，前 3 次进行判断-加法-移位操作、最后 1 次进行判断-加法操作，运算过程如下表所示：

循环次数	部分积高位	乘数 (及部分积低位)	T	说明
4	000.000000	0 0 1 0 0 1 1 0	0	初始部分积 P_0=000、T=0
	+000.110110			b_1b_0T=100，应+[2|A|]$_{补}$、T←0
	000.110110			
3	000.001101	1 0 0 0 1 0 0 1	0	算术右移 2 位(最高符号位为真符号位)
	+000.011011			b_1b_0T=010，应+[|A|]$_{补}$、T←0
	000.101000			
2	000.001010	0 0 1 0 0 0 1 0	0	算术右移 2 位
	+000.110110			b_1b_0T=100，应+[2|A|]$_{补}$、T←0
	001.000000			
1	000.010000	0 0 0 0 1 0 0 0	0	算术右移 2 位
	+000.000000			b_1b_0T=000，应+0
0	000.010000	0 0 0 0 1 0		不移位，乘积数值部分为 0.0100000000010

即|A|×|B|=0.010000000010，故[A×B]$_{原}$=1.010000000010。

28. 对下列 A 和 B，请用补码一位乘法(Booth 算法)求 A×B。

（1）A=0.110111，B=-0.101010；　（2）A=19，B=35

解：（1）由题意[A]$_{补}$=0.110111，[B]$_{补}$=1.010110，[-A]$_{补}$=1.001001，

因连同符号一起运算，故共循环 7 次，进行判断-加法-移位操作(最后 1 次不移位)。运算过程如下表所示：

循环	部分积	乘数	附加位	操作说明
7	0.000000	1 0 1 0 1 1 0	0	初始[P_0]$_{补}$=0，b_{-n}=0
	+0.000000			$b_{-n-1}b_{-n}$=00，部分积+0
	0.000000			
6	0.000000	0 1 0 1 0 1 1	0	部分积和乘数同时算术右移 1 位
	+1.001001			$b_{-n-1}b_{-n}$=10，部分积+[-A]$_{补}$
	1.001001			
5	1.100100	1 0 1 0 1 0 1	1	部分积和乘数同时算术右移 1 位
	+0.000000			$b_{-n-1}b_{-n}$=11，部分积+0
	1.100100			
4	1.110010	0 1 0 1 0 1 0	1	部分积和乘数同时算术右移 1 位
	+0.110111			$b_{-n-1}b_{-n}$=01，部分积+[A]$_{补}$
	0.101001			
3	0.010100	1 0 1 0 1 0 1	0	部分积和乘数同时算术右移 1 位
	+1.001001			$b_{-n-1}b_{-n}$=10，部分积+[-A]$_{补}$
	1.011101			
2	1.101110	1 1 0 1 0 1 0	1	部分积和乘数同时算术右移 1 位
	+0.110111			$b_{-n-1}b_{-n}$=01，部分积+[A]$_{补}$
	0.100101			
1	0.010010	1 1 1 0 1 0 1	0	部分积和乘数同时算术右移 1 位
	+1.001001			$b_{-n-1}b_{-n}$=10，部分积+[-A]$_{补}$
	1.011011			
0	1.011011	1 1 1 0 1 0		最后一次不右移

故[A×B]$_{补}$=1.011011111010。

（2）由题意[A]$_{补}$=0010011，[B]$_{补}$=0100011，[-A]$_{补}$=1101101，

共循环 7 次，进行判断-加法-移位操作(最后 1 次不移位)。运算过程如下表所示：

循环	部分积	乘数	附加位	操作说明
7	0000000	0 1 0 0 0 1 1	0	初始$[P_0]_{补}=0$，$b_{-n}=0$
	+1101101			$b_{-n-1}b_{-n}=10$，部分积+$[-A]_{补}$
	1101101			
6	1110110	1 0 1 0 0 0 1	1	部分积和乘数同时算术右移 1 位
	+0000000			$b_{-n-1}b_{-n}=11$，部分积+0
	1110110			
5	1111011	0 1 0 1 0 0 0	1	部分积和乘数同时算术右移 1 位
	+0010011			$b_{-n-1}b_{-n}=01$，部分积+$[A]_{补}$
	0001110			
4	0000111	0 0 1 0 1 0 0	0	部分积和乘数同时算术右移 1 位
	+0000000			$b_{-n-1}b_{-n}=00$，部分积+0
	0000111			
3	0000011	1 0 0 1 0 1 0	0	部分积和乘数同时算术右移 1 位
	+0000000			$b_{-n-1}b_{-n}=00$，部分积+0
	0000011			
2	0000001	1 1 0 0 1 0 1	0	部分积和乘数同时算术右移 1 位
	+1101101			$b_{-n-1}b_{-n}=10$，部分积+$[-A]_{补}$
	1101110			
1	1110111	0 1 1 0 0 1 0	1	部分积和乘数同时算术右移 1 位
	+0010011			$b_{-n-1}b_{-n}=01$，部分积+$[A]_{补}$
	0001010			
0	0001010	0 1 1 0 0 1		最后一次不右移

故$[A\times B]_{补}=0001010011001$。

29. 对下列 A 和 B，请用不恢复余数法(交替加减法)原码除法求 $A\div B$。

（1）$A=0.100111$，$B=-0.101010$；　（2）$A=0100100110$，$B=10101$

解：（1）由题意$[A]_{原}=0.100111$，$[B]_{原}=1.101010$，

$[|A|]_{补}=0.100111$，$[|B|]_{补}=0.101010$，$[-|B|]_{补}=1.010110$，

不恢复余数法原码除法求解过程如下表所示：

被除数(余数)	商	说明
0.100111	* * * * * * *	被除数$=[|A|]_{补}$，商＝任意值
+1.010110		减去除数，+$[-|B|]_{补}$
1.111101	0.	余数为负，除法不溢出，上商为 0
1.111010	0.	余数与部分商同时逻辑左移 1 位
+0.101010		因 $q_0=0$(即余数为负)：加上除数，+$[|B|]_{补}$
0.100100	0.1	余数为正，上商为 1
1.001000	0.1	余数与部分商同时逻辑左移 1 位
+1.010110		因 $q_1=1$(即余数为正)：减去除数，+$[-|B|]_{补}$
0.011110	0.1 1	余数为正，上商为 1
0.111100	0.1 1	余数与部分商同时逻辑左移 1 位
+1.010110		因 $q_2=1$(即余数为正)：减去除数，+$[-|B|]_{补}$
0.010010	0.1 1 1	余数为正，上商为 1
0.100100	0.1 1 1	余数与部分商同时逻辑左移 1 位
+1.010110		因 $q_3=1$(即余数为正)：减去除数，+$[-|B|]_{补}$
1.111010	0.1 1 1 0	余数为负，上商为 0
1.110100	0.1 1 1 0	余数与部分商同时逻辑左移 1 位
+0.101010		因 $q_4=0$（即余数为负）：加上除数，+$[|B|]_{补}$
0.011110	0.1 1 1 0 1	余数为正，上商为 1
0.111100	0.1 1 1 0 1	余数与部分商同时逻辑左移 1 位
+1.010110		因 $q_5=1$（即余数为负）：减去除数，+$[-|B|]_{补}$
0.010010	0.1 1 1 0 1 1	余数为正，上商为 1，
		因 $q_6=1$，余数不用修正

所以|商|＝0.$q_1\,q_2\,q_3\,q_4\,q_5\,q_6$＝0.111011，|余数|=0.010010。

由于商符 s_Q＝1⊕0＝1，故$[A/B]_{原}$＝1.111011，$[A\%B]_{原}$＝0.010010，

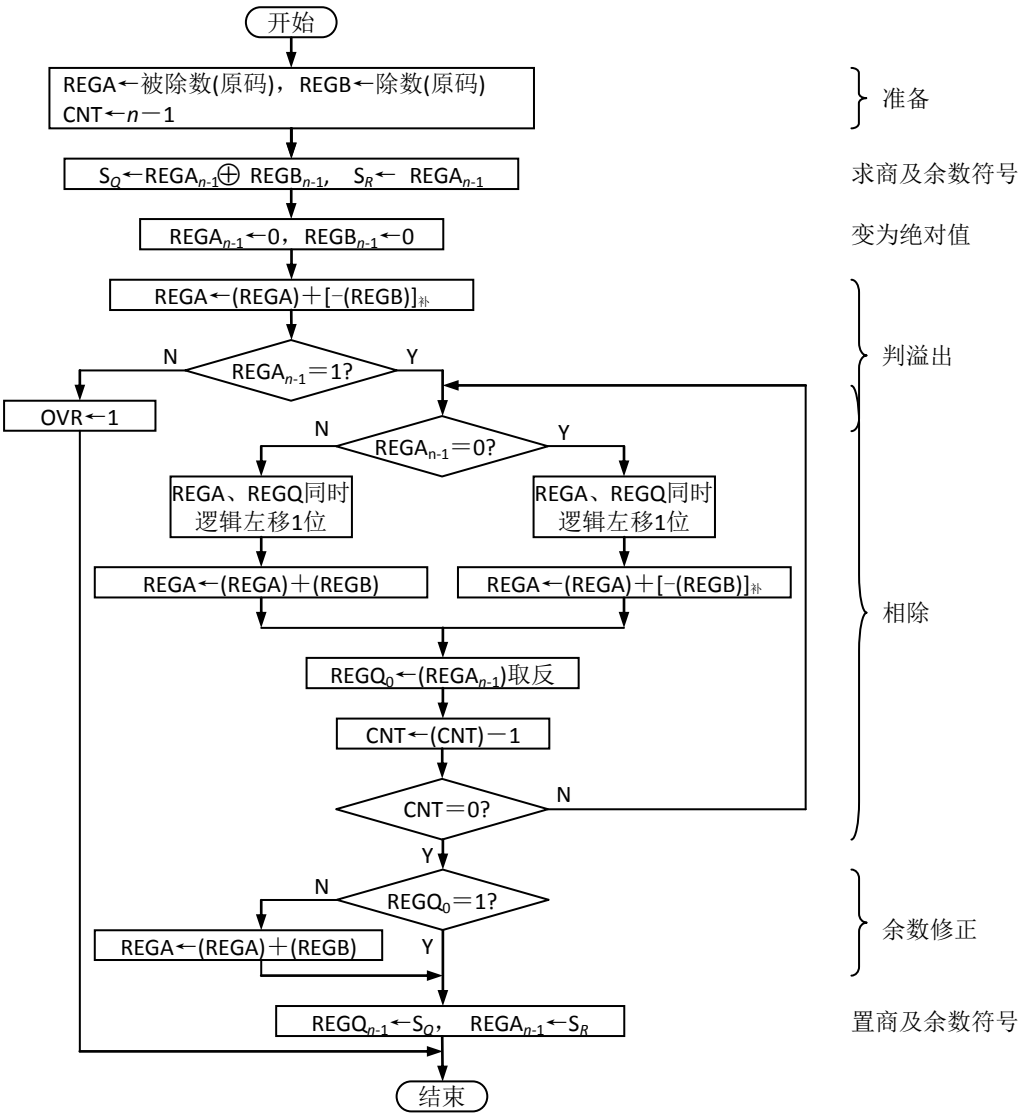
A/B＝−0.111011，$A\%B$＝+0.010010。

30. 对于不恢复余数法原码除法而言，图 2.16 的运算流程并不完全适用于整数除法，因为整数除法开始时，被除数存放在 REGA 及 REGQ 中，请将图 2.16 改成适合整数除法的流程图(提示：应先移位、后上商)。

解：图 2.16 的原码除法中，

“判溢出”阶段的“$REGQ_0$←0”对应的是最后结果的符号位，可省略(最后被覆盖)；

“相除”阶段的“$REGQ_0$=1？”可用“$REGA_{n-1}$=0？”代替。



31. 若 A＝−0.10101，B＝0.11011，请用不恢复余数法补码除法求 $A\div B$。

解：由题意$[A]_{补}$＝1.01011,$[B]_{补}$＝0.11011,$[-B]_{补}$＝1.00101，

不恢复余数补码除法过程如下表所示：

被除数(余数)	商$[Q]_{反}$	说明
1.01011	* * * * * *	$[A]_{补}$与$[B]_{补}$异号
+0.11011		求$[R_0']_{补}$时+$[B]_{补}$
0.00110	1.	$[R_0']_{补}$与$[B]_{补}$同号，不溢出、q_0←1
0.01100	1.	余数与部分商同时左移 1 位
+1.00101		求新余数时+$[-B]_{补}$
1.10001	1.0	$[R_1']_{补}$与$[B]_{补}$异号，q_1←0
1.00010	1.0	余数与部分商同时左移 1 位
+0.11011		求新余数时+$[B]_{补}$
1.11101	1.0 0	$[R_2']_{补}$与$[B]_{补}$异号，q_2←0
1.11010	1.0 0	余数与部分商同时左移 1 位
+0.11011		求新余数时+$[B]_{补}$
0.10101	1.0 0 1	$[R_3']_{补}$与$[B]_{补}$同号，q_3←1
1.01010	1.0 0 1	余数与部分商同时左移 1 位
+1.00101		求新余数时+$[-B]_{补}$
0.01111	1.0 0 1 1	$[R_4']_{补}$与$[B]_{补}$同号，q_4←1
0.11110	1.0 0 1 1	余数与部分商同时左移 1 位
+1.00101		求新余数时+$[-B]_{补}$
0.00011	1.0 0 1 1 1	$[R_5']_{补}$与$[B]_{补}$同号，q_5←1
+1.00101		q_0=1、q_5=1，即不够减，修正余数+$[-B]_{补}$
1.01000		

所以，$[Q]_{反}$=1.00111，$[Q]_{补}$=1.00111+0.00001=1.01000，$[R]_{补}$=1.01000。

32. 若浮点数用 4 位阶码(含 1 位阶符)、7 位尾数(含 1 位数符)格式表示，阶码及尾数均为补码，请用浮点运算方法计算 $A+B$。

（1）$A=2^{-011}\times 0.101100$，$B=2^{-010}\times(-0.011100)$；

（2）$A=2^{101}\times(-0.100101)$，$B=2^{100}\times(-0.001111)$

解：假设运算时采用 2 位警戒位、双符号位和舍入法。

（1）由题意$[A]_{浮}$=1101 0101100，$[B]_{浮}$=1110 1100100，

①对阶，$[\Delta E]_{补}$=11101−11110=11111，ΔE=-1，A的尾数右移 1 位、阶码加 1，

即$[M_A']_{补}$=00.01011<u>00</u>，$[M_B']_{补}$=11.10010<u>00</u>，$[E_A']_{补}=[E_B']_{补}$=11110

②尾数加减，$[M_A']_{补}+[M_B']_{补}$=00.01011<u>00</u>+11.10010<u>00</u>=11.11010<u>00</u>，

即$[A+B]_{浮}$=11110 1111101<u>00</u>

③规格化，尾数不溢出、(补码)符号位与最高数值位相同，左规 3 次后，

得$[A+B]_{浮}$=11011 1101000<u>00</u>

④尾数舍入，警戒位为 00，选择“舍”(即“不舍不入”)，

即$[A+B]_{浮}$=11011 11010000

⑤溢出判断，阶码符号为 11，$A+B$ 结果不溢出。

故$[A+B]_{浮}$=1011 1010000，$A+B=2^{-101}\times(-0.110000)$。

（2）由题意$[A]_{浮}$=0101 1011011，$[B]_{浮}$=0100 1110001，

①对阶，$[\Delta E]_{补}$=00101−00100=00001，ΔE=1，B的尾数右移 1 位、阶码加 1，

即$[M_A']_{补}$=11.01101<u>00</u>，$[M_B']_{补}$=11.11100<u>10</u>，$[E_A']_{补}=[E_B']_{补}$=00101

②尾数加减，$[M_A']_{补}+[M_B']_{补}$=11.01101<u>00</u>+11.11100<u>10</u>=11.01001<u>10</u>，

即$[A+B]_{浮}$=00101 1101001<u>10</u>

③规格化，尾数不溢出、(补码)符号位与最高数值位不同，尾数无需规格化，

即$[A+B]_{浮}$=00101 1101001<u>10</u>

④尾数舍入，由于尾数为负、警戒位为 10，选择“舍”，

即$[A+B]_{浮}$=00101 11010011

⑤溢出判断，阶码符号为 00，$A+B$ 结果不溢出。

故$[A+B]_{浮}$=0101 1010011，$A+B=2^{+101}\times(-0.101101)$。

33. 若浮点数表示格式为：5 位阶码(含 1 位阶符)、8 位尾数(含 1 位数符)，阶码用移码表示、尾数用补码表示，浮点运算时采用双符号位运算、警戒位为 3 位、采用舍入法，请用浮点运算方法计算下列各题，并写出运算结果的机器码。

（1）$[2^{15}\times{}^{11}\!/_{16}]+[2^{13}\times(-{}^{9}\!/_{16})]$；

（2）$[2^{-13}\times{}^{13}\!/_{16}]+[2^{-14}\times(-{}^{5}\!/_{8})]$

解：（1）由题意$[A]_{浮}$=11111 01011000，$[B]_{浮}$=11101 10111000，

①对阶，$[E_B]_{补}$=01101，$[-E_B]_{补}$=10011，

$[\Delta E]_{移}=[E_A]_{移}-[E_B]_{补}=[E_A]_{移}+[-E_B]_{补}$=111111+110011=110010，

ΔE=2，B的尾数右移 2 位、阶码加 2，

即$[M_A']_{补}$=00.1011000000，$[M_B']_{补}$=11.1101110000，$[E_A']_{移}=[E_B']_{移}$=111111

②尾数加减，$[M_A']_{补}+[M_B']_{补}$=00.1011000000+11.1101110000=00.1000110000，

即$[A+B]_{浮}$=111111 001000110000

③规格化，尾数不溢出、(补码)符号位与最高数值位不同，尾数无需规格化，

即$[A+B]_{浮}$=111111 001000110000

④尾数舍入，由于警戒位为 000，选择“舍”（即“不舍不入”），

即$[A+B]_{浮}$=111111 001000110

⑤溢出判断，阶码符号位为 11，$A+B$ 结果不溢出。

故$[A+B]_{浮}$=11111 01000110。

（2）由题意$[A]_{浮}$=00011 01101000，$[B]_{浮}$=00010 10110000，

①对阶，$[\Delta E]_{移}=[E_A]_{移}-[E_B]_{补}=[E_A]_{移}+[-E_B]_{补}$=000011+001110=010001，

ΔE=1，B的尾数右移 1 位、阶码加 1，

即$[M_A']_{补}$=00.1101000000，$[M_B']_{补}$=11.1011000000，$[E_A']_{补}=[E_B']_{补}$=000011

②尾数加减，$[M_A']_{补}+[M_B']_{补}$=00.1101000000+11.1011000000=00.1000000000，

即$[A+B]_{浮}$=000011 001000000000

③规格化，尾数不溢出、(补码)符号位与最高数值位不同，尾数无需规格化，

即$[A+B]_{浮}$=000011 001000000000

④尾数舍入，由于警戒位为 000，选择“舍”（即“不舍不入”），

即$[A+B]_{浮}$=000011 001000000

⑤溢出判断，阶码符号位为 00，$A+B$ 结果不溢出。

故$[A+B]_{浮}$=00011 01000000。

34. 若浮点数表示格式为：6 位阶码(含 1 位阶符)、10 位尾数(含 1 位数符)，阶码用原码表示、尾数用补码表示，浮点乘法结果保留 1 倍长度，请用浮点乘法求$[2^{-13}\times{}^{101}\!/_{128}]\times[2^{-4}\times(-{}^{135}\!/_{256})]$。

解：由题意，$[A]_{浮}$=101101 0110010100，$[B]_{浮}$=100100 1011110010，

①阶码运算，$[E_A]_{原}$=101101、$[E_B]_{原}$=100100，则$[E_A]_{补}$=110011、$[E_B]_{补}$=111100，

$[E_A+E_B]_{补}=[E_A]_{补}+[E_B]_{补}$=110011+111100=101111，$[E_A+E_B]_{原}$=110001；

②尾数相乘，$[M_A]_{补}$=0.110010100，$[M_B]_{补}$=1.011110010，$[-M_A]_{补}$=1.001101100

采用 Booth 算法相乘，得$[M_A\times M_B]_{补}$=1.100101010111101000，

即$[A\times B]_{浮}$=110001 1100101010111101000

③规格化，因符号位与最高数值位相同，需左规格化，左规 1 次后，

即$[A\times B]_{浮}$＝110010 1001010101111010000

④舍入处理，由于尾数保留 1 倍长度，需进行舍入处理，

因尾数为负，按负数补码的舍入规则，选择“入”，

即$[A\times B]_{浮}$＝110010 1001010110，$A\times B=2^{-18}\times(-0.110101010)$。

35. 对图 2.22 的先行进位电路进行修改，形成一个便于级联的先行进位电路，即输入端为 G_3～G_0、P_3～P_0 及 C_{-1}，输出端为 C_2～C_0、G 及 P。其中 G 及 P 实现的逻辑是该进位电路的进位产生函数和进位传递函数。

解：由于图 2.22 的先行进位电路没有进位产生函数和进位传递函数信号，故多个这种电路间只能实现串行进位、不能实现先行进位；

先行进位电路的改进主要是增加进位产生函数信号 G 和进位传递函数信号 P 的逻辑。

根据先行进位逻辑可知：

$C_0=G_0+P_0C_{-1}$

$C_1=G_1+P_1C_0=G_1+P_1G_0+P_1P_0C_{-1}$

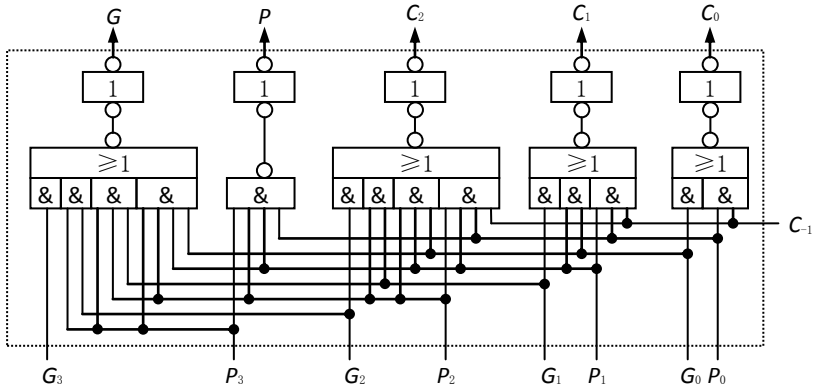
$C_2=G_2+P_2C_1=G_2+P_2G_1+P_2P_1G_0+P_2P_1P_0C_{-1}$

$C_3=G_3+P_3C_2=G_3+P_3G_2+P_3P_2G_1+P_3P_2P_1G_0+P_3P_2P_1P_0C_{-1}$

若 $C_3=G+PC_{-1}$，则 $G=G_3+P_3G_2+P_3P_2G_1+P_3P_2P_1G_0$，

$P=P_3P_2P_1P_0$

新改进的先行进位电路的 C_2～C_0、G 及 P 的逻辑如下图所示：

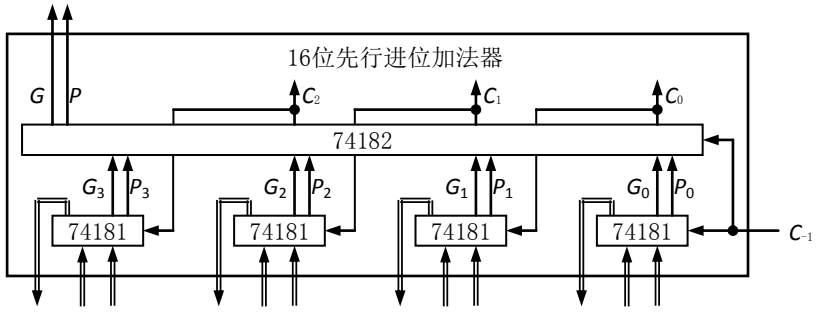


36. 试用 74181、图 2.22 的先行进位电路及题 35 的先行进位电路构建一个 64 位并行 ALU，画出各部件或电路间的连接图。

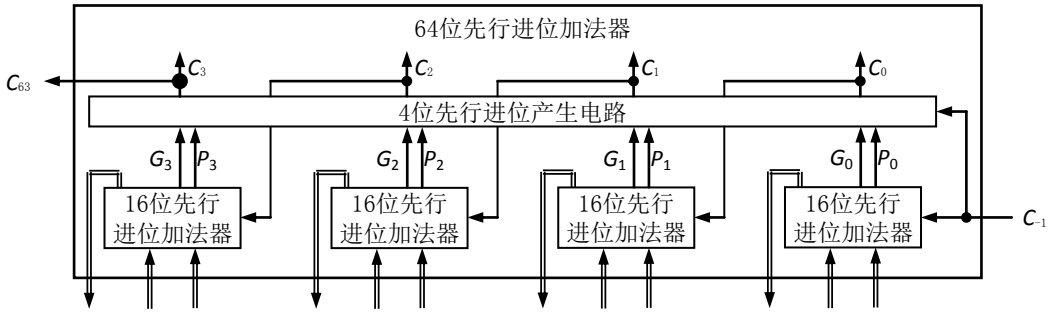
解：题 35 改进的先行进位电路与 74182 芯片完全相同，本题记为 74182。

64 位并行 ALU 可由 4 个 16 位先行进位加法器、采用先行进位逻辑电路连接而成，而 16 位先行进位加法器可由 4 个 74181、1 个 74182 连接而成。

（1）16 位先行进位加法器的逻辑如下图所示：



（2）64 位并行加法器的逻辑如下图所示：



37. 图 2.28 中 29C101 输出状态信号 C 及 OVR，用于表示无符号及有符号运算结果是否溢出，这很正常。相对于只输出一个溢出信号，输出两个信号有什么好处？29C101 还输出 F_{15} 及 F=0 两个状态信号，有什么功效？

答：（1）由于有/无符号运算的溢出逻辑不同，因此，ALU 的功能选择信号必须能够区分是有符号运算、还是无符号运算；

而 29C101 同时输出 C 及 OVR 两个信号，使得功能选择信号不需要区分有/无符号运算，减少了功能选择信号线的数量，而运算是否溢出由软件根据需要选择 C 或 OVR 进行判断。

（2）由于 CPU 中的关系运算通常用算术运算(减法)和逻辑运算实现，而 ALU 刚好能实现算术运算和逻辑运算；

而关系运算的结果可以用算术运算(减法)的“结果是否为零”、“结果是否为负”状态组合表示，如结果为零是表示 A＝B、结果为负时表示 A＜B；

而 F=0 及 F_{15} 这两个信号刚好可以表示“结果是否为零”、“结果是否为负”的状态，因此，F_{15} 及 F=0 这两个状态信号有效地支持了基于 ALU 的关系运算实现。

第 3 章

1. 解释概念或术语：RAM、SAM、DAM、ROM，主存、辅存、存储元、存储单元、存储字、存储阵列、二维译码。

答：略。

2. 说明存取时间、存储（存取）周期和带宽的区别。

答：存取时间(T_A)指存储器从接收到操作命令到完成操作(信息被送到数据引脚上或引脚上信息被存到存储元中)的时间；

存取周期(T_M)指存储器完成访问并恢复到就绪状态的全部时间，即存储器连续两次完成访问的最短间隔时间；

带宽(B_M)指存储器单位时间内可以提供数据的最大速率，即单位时间内最多可读写的二进制位数；

由此可见，T_A 及 T_M 反映存储器完成访问的延迟，与数据线数量无关，而 B_M 反映存储器传输数据的速度，与数据线数量有关；T_M 比 T_A 多了恢复到就绪状态时间，因此 $T_M \geqslant T_A$。

3. 层次结构存储系统能够满足用户大容量、高速度、低价格的原因是什么？不同层存储器中的内容有什么关系？CPU 按哪种存储器的地址访问存储系统？

答：假设层次结构存储系统由 Cache、主存、辅存三种存储器组成。

（1）若将近期常用信息放在高速度的 Cache 中，可满足用户高速度需求，而程序访问的局部性规律使得将近期常用信息放在 Cache 和主存中成为可能；由于所有信息最终均存放在辅存中，若使辅存容量较大，自然满足了用户大容量需求；若使每位价格 $C_{\text{Cache}} > C_{\text{主存}} > C_{\text{辅存}}$、容量 $S_{\text{Cache}} << S_{\text{主存}} << S_{\text{辅存}}$， 则$(C_{\text{Cache}}S_{\text{Cache}} + C_{\text{主存}}S_{\text{主存}} + C_{\text{辅存}}S_{\text{辅存}}) \div (S_{\text{Cache}} + S_{\text{主存}} + S_{\text{辅存}}) \approx C_{\text{辅存}}$，保持了低价格的特点。

（2）前方存储器(离 CPU 近些)中信息为后方存储器(离 CPU 远些)中信息的副本。因为存储系统采用层次结构，而所有信息最初及最终均存放在辅存中。

（3）现代计算机中，主存是 CPU 默认访问的存储器，例如欲执行的指令和数据必须预先调入主存，CPU 方可执行。而层次结构存储系统的目标仅仅是优化主存的性能及性能/价格，因此，CPU 按主存地址访问存储系统。

4. 六管 MOS 型 SRAM 存储元存储、读、写信息的原理是什么？各种存储器芯片为什么常采用二维译码方式？SRAM 芯片为什么要求地址引脚信号先于 $\overline{\text{CS}}$ 建立、后于 $\overline{\text{CS}}$ 撤销？

答：（1）六管 MOS 型 SRAM 存储元的核心是 T_1～T_4 构成的触发器电路，T_5、T_6 的作用是操作开关、控制是否对该存储元进行操作。

对信息写入而言，需两个步骤实现：①在字选择线上加高电位(保持到写操作完成时)，使 T_5、T_6 在写操作过程中均处于导通状态；②写“0”时，在 D 线上加低电位、$\overline{\text{D}}$ 线上加高电位，则 T_1 管导通、T_2 管截止，达到了“0”的状态，写“1”时，在 D 线上加高电位、$\overline{\text{D}}$ 线上加低电位，则 T_1 管截止、T_2 管导通，达到了“1”的状态。

对信息存储而言(T_5、T_6 管处于截止状态)，由于触发器的状态自锁电路可稳定地保持 T_1、T_2 管的状态不变，因此，只要不掉电，存储元可保持所写入的信息永远不变。

对信息读出而言，同样需两个步骤实现：①在字选择线上加高电位(保持到读操作完成时)，使 T_5、T_6 在读操作过程中均处于导通状态；②若存储元信息是“0”，则 D 线上电位下降、$\overline{\text{D}}$ 线上电位上升，用连接 D 线与 $\overline{\text{D}}$ 线的差动放大器可以输出所存信息是“0”还是“1”。

（2）因为存储器常将所有存储元排列成二维矩阵形式，这样可以使存储元间连线长

度最短、线路信号延迟最小；而采用二维译码方式可以有效地支持存储元二维排列方式的实现：减少连线数量、减少连线长度；

（3）由于$\overline{CS}$有效时，SRAM 内部的读/写控制电路根据$\overline{WE}$引脚信号立即产生读或写操作；若$\overline{CS}$有效时地址引脚信号尚未稳定，则将先对未知存储单元、再对目标存储单元进行读或写操作，读操作不会产生危害，写操作将会产生不可挽回的后果；因此，SRAM 芯片要求地址引脚信号先于$\overline{CS}$建立、后于$\overline{CS}$撤销。

5. 某容量为 16K×16bit 的 SRAM 芯片，其地址引脚（线）、数据引脚（线）各是多少？若某 SRAM 芯片容量为 1K 个存储元，地址引脚有 8 根，则其数据引脚有多少根？

解：（1）芯片的地址引脚有 $\log_2(16\times2^{10})=14$ 根，数据引脚有 16 根；

（2）数据引脚有$(1\times2^{10})/2^8=4$ 根。

6. 若用 16K×4bit SRAM 芯片构成 16K×16bit 的 SRAM，问：

（1）需要 SRAM 芯片多少片？

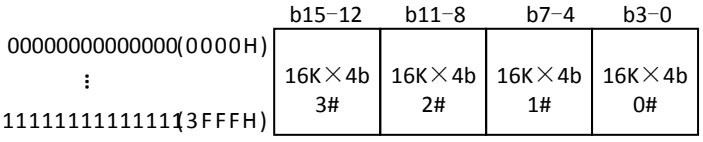
（2）各芯片在所组成 SRAM 存储空间中的位置？

（3）各芯片片选线的有效逻辑是什么？

（4）画出所组成 SRAM 的信号线与内部各芯片引脚的连接图。

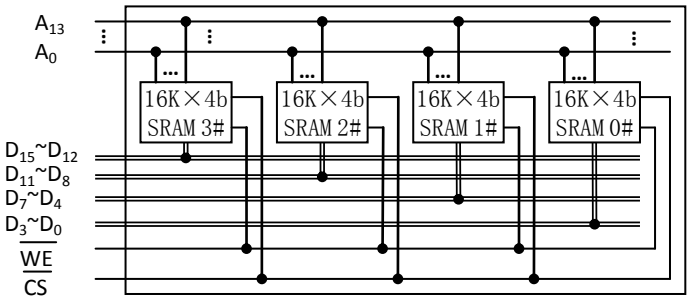
解：（1）需要 SRAM 芯片(16K/16K)×(16bit/4bit)＝4 片

（2）各芯片在 SRAM 存储空间中的位置如下图：



（3）各芯片的片选线有效逻辑为：$\overline{CS}_0=\overline{CS}_1=\overline{CS}_2=\overline{CS}_3=\overline{CS}$；

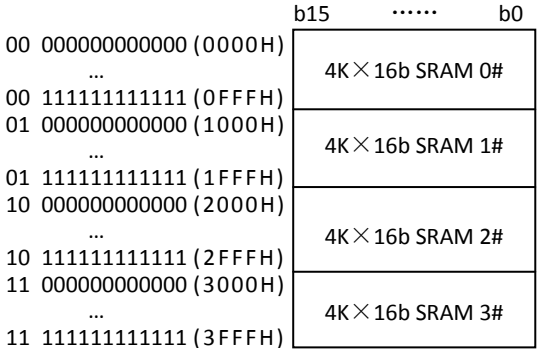
（4）SRAM 的信号线与内部各芯片引脚的连接如下图：



7. 若用 4K×16bit SRAM 芯片构成 16K×16bit SRAM，回答与题 6 相同的问题。

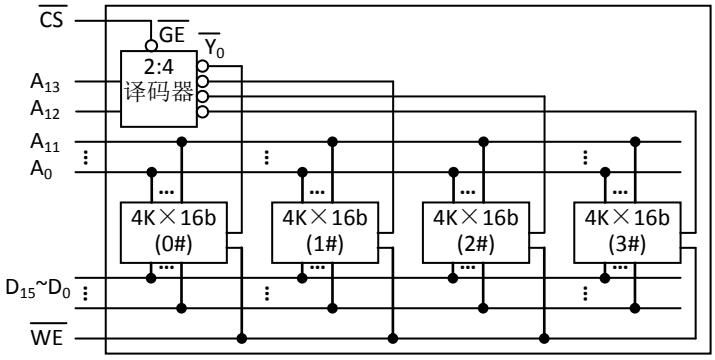
解：（1）需要 SRAM 芯片(16K/4K)×(16bit/16bit)＝4 片

（2）各芯片在 SRAM 存储空间中的位置如下图：



（3）各芯片的片选线有效逻辑为：$\overline{CS}_0=\overline{CS}\,\&\,\overline{A_{13}}\,\&\,\overline{A_{12}}$、$\overline{CS}_1=\overline{CS}\,\&\,\overline{A_{13}}\&A_{12}$、
$\overline{CS}_2=\overline{CS}\,\&\,A_{13}\&\,\overline{A_{12}}$、$\overline{CS}_3=\overline{CS}\,\&\,A_{13}\&A_{12}$

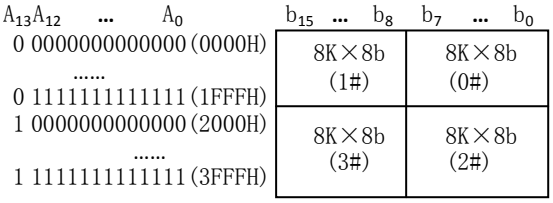
（4）SRAM 的信号线与内部各芯片引脚的连接如下图：



8. 若用 8K×8bit SRAM 芯片构成 16K×16bit SRAM，回答与题 6 相同的问题。

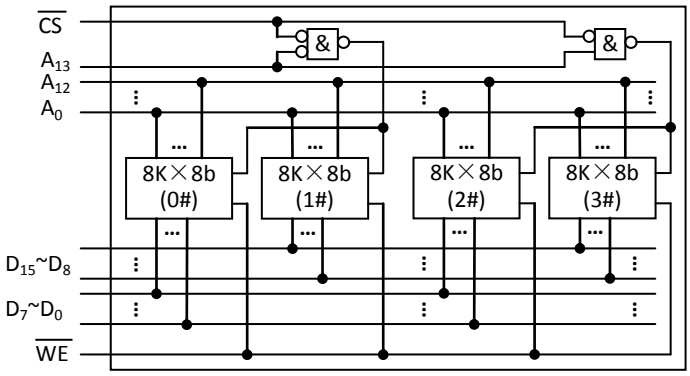
解：（1）需要 SRAM 芯片(16K/8K)×(16bit/8bit)＝4 片

（2）各芯片在 SRAM 存储空间中的位置如下图：



（3）各芯片的片选线有效逻辑为：$\overline{CS}_0=\overline{CS}_1=\overline{CS}\,\&\overline{A_{13}}$、$\overline{CS}_2=\overline{CS}_3=\overline{CS}\,\&A_{13}$

（4）SRAM 的信号线与内部各芯片引脚的连接如下图：



9. 对单管 MOS 型 DRAM 芯片而言，存储元存储、读、写信息的原理是什么？读 DRAM 存储元比读 SRAM 存储元慢的原因有哪些？行刷新可以提高刷新效率，其对 DRAM 芯片的要求是什么？采用地址分两次传送方式有什么好处？

答：（1）单管 MOS 型 DRAM 存储元的信息存储用电容 C_S 实现，T_1 管用作操作开关、控制是否对该存储元进行操作；

对信息写入而言，需两个步骤：①在字选择线上加高电位，使 T_1 导通；②在数据线 D 上加所写数据，C_S 将随所写数据而被充电或放电，信息被写入。

对信息存储而言(T_1 管截止)，C_S 中信息因无放电回路而得以保持，但因 C_S 会缓慢泄漏电荷，需定时进行刷新操作以长久保持信息。

对信息读出而言，需三个步骤：①将 D 线预充电为高电位，通过 C_D 实现；②在字选择线上加高电位，使 T_1 导通，D 线上电位将随 C_S 中有无电荷而产生变化（C_S 无电荷时，D 线上电位下降，否则 D 上电位基本不变），放大该电位变化可读出所存储信息；③立即进行数

据再生操作（用所读出信息重新写入）。

（2）读 DRAM 存储元较慢的原因主要是因为 DRAM 增加了三方面的操作：

①读信息前需对 C_D 与充电，②读信息后需再生信息，③存储元需定时刷新。

（3）由于 DRAM 芯片内部通常采用二维译码方式、各列存储元通过列译码器输出与 I/O 电路连接，行刷新指同一行所有存储元同时进行刷新操作，因此，要求 DRAM 芯片控制电路中增加刷新控制机构(区分读/写与刷新操作)，每列增加存储元刷新部件(如刷新放大器等刷新操作功能部件)，列译码器所有输出无效(即列地址刷新时无用、实现同时刷新功能)。

（4）可以减少地址引脚数量，进一步减小芯片封装面积，用$\overline{\text{RAS}}$、$\overline{\text{CAS}}$两个信号来区分当前地址信号是行地址还是列地址；可以不增加控制引脚数量，同样完成所有功能，使$\overline{\text{RAS}}$信号在整个操作过程中均有效，则$\overline{\text{RAS}}$引脚可代替片选引脚$\overline{\text{CS}}$，使$\overline{\text{CAS}}$信号在刷新操作时一直无效(读/写操作时适时有效)，则无需增加刷新命令引脚。

10. 某 DRAM 芯片的存储阵列有 1024 行、存取时间为 0.5μs、存储元的最大刷新周期为 2ms，若 CPU 在 1μs 内至少要访问一次，请选择你认为较合理的芯片刷新方式，并说明理由。两次刷新的最大时间间隔是多少？芯片刷新一遍共花费多长时间。

解：（1）由于 DRAM 最大访问频率＝1/(0.5μs)＝2 次/μs，而 CPU 访存频率≥1 次/μs，

采用分散式刷新时，CPU 访存频率≤2000μs/(0.5μs+0.5μs)＝2000 次/2ms，≤1 次/μs；

采用集中式刷新时，CPU 访存频率≤(2000μs/0.5μs－1024)＝2976 次/2ms，≥1 次/μs，其中有 1024×0.5μs＝512μs 的 CPU 访存操作延迟巨大(在死区中)；

采用异步式刷新时，CPU 访存频率≤2976 次/2ms，≥1 次/μs，其中每 2000μs/1024＝1.95μs 中，超过[(2000/1024－0.5μs)/ 0.5μs]＝2.9 次的 CPU 访存操作延迟增加；

三种刷新方式中，分散式刷新不符合题目要求，最合理的芯片刷新方式是异步式刷新。

（2）采用异步式刷新时，两次刷新最大的时间间隔为 2000μs/1024=1.95μs。

（3）芯片刷新一遍所需时间为 1024×0.5μs=512μs。

11. 现分别用 16K×4bit、4K×16bit 地址分两次传送的 DRAM 芯片构成 16K×16bit 的 DRAM，请分别回答下列问题：

（1）需要 DRAM 芯片多少片？

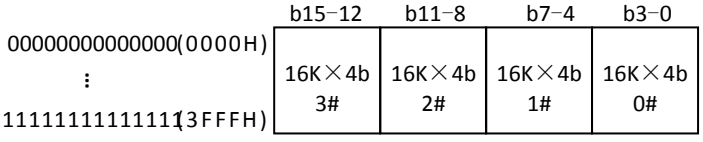
（2）各芯片在所组成 DRAM 存储空间中的位置？

（3）画出所组成 DRAM 的信号线与内部各芯片引脚的连接图。

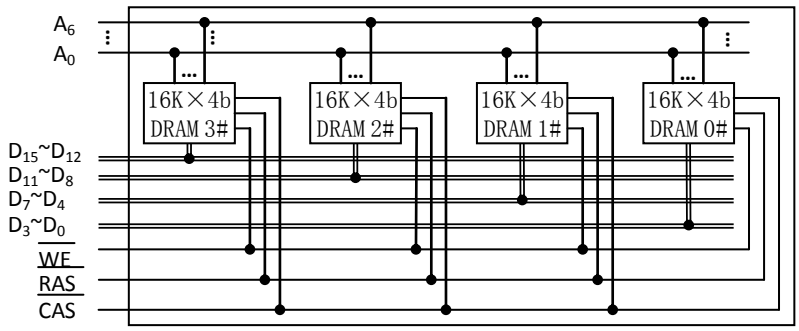
解：当用 16K×4bit 的 DRAM 芯片构成 16K×16bit 的 DRAM 时，

（1）需要 DRAM 芯片数＝(16K/16K)×(16bit/4bit)＝4 片 16K×4bit 芯片；

（2）各芯片在 DRAM 存储空间中的位置如下图：



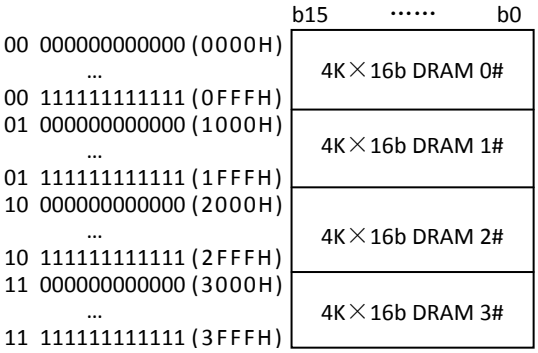
（3）所组成 DRAM 的信号线与内部各芯片引脚的连接如下图：



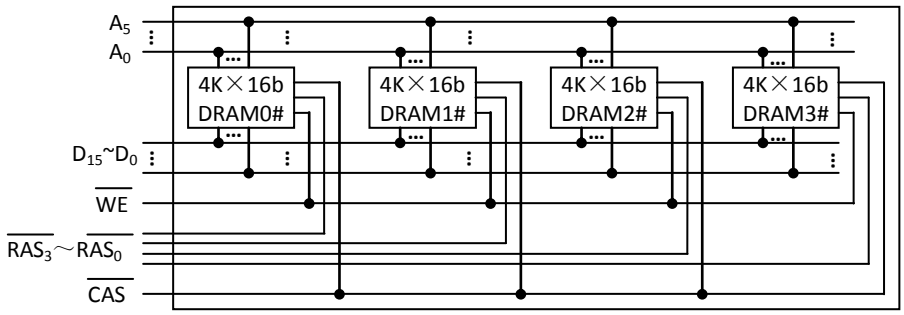
当用 4K×16bit 的 DRAM 芯片构成 16K×16bit 的 DRAM 时，

（1）需要 DRAM 芯片数＝(16K/4K)×(16bit/16bit)＝4 片 4K×16bit 芯片；

（2）各芯片在 DRAM 存储空间中的位置如下图：



（3）所组成 DRAM 的信号线与内部各芯片引脚的连接如下图：



12. 某计算机中，CPU 有 20 根地址引脚（A_{19}～A_0）、8 根数据引脚（D_7～D_0），控制引脚由 $\overline{ADS}$、IO/$\overline{M}$ 及 R/$\overline{W}$ 三根引脚组成，若主存配置为 64KB 的 ROM 及 192KB 的 SRAM，ROM 在主存空间的低端。现有 64K×4 位 ROM 芯片、32K×8 位 SRAM 芯片若干。

（1）主存的地址线、数据线各有多少位？各芯片在主存空间中的位置？

（2）相对于主存，各芯片片选线的有效逻辑是什么？画出主存信号线与内部各芯片引脚的连接图。

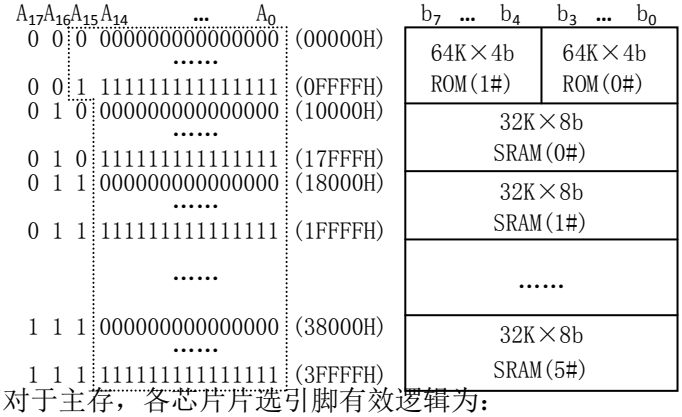
（3）相对于 CPU，主存被选中的条件（主存有效的逻辑）是什么？画出 CPU 与主存内部各芯片引脚的连接图。

解：（1）主存总空间为 64KB＋192KB＝256KB，由于 CPU 的数据引脚宽度为 8 位，即主存单元长度为 8 位，故主存的数据线有 8 位，地址线有 $\log_2(256KB/1B)$＝18 位；

主存 ROM 空间需 ROM 芯片(64K/64K)×(8b/4b)＝2 片，

主存 RAM 空间需 SRAM 芯片(192K/32K)×(8b/8b)＝6 片；

各芯片在主存空间中的位置如下图：



（2）相对于主存，各芯片片选引脚有效逻辑为：

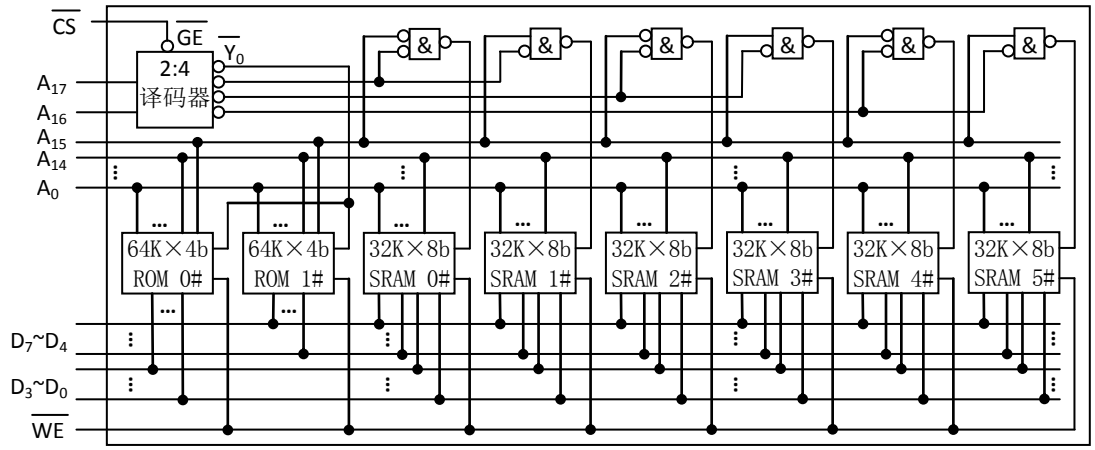
$\overline{CS}_{ROM0}=\overline{CS}_{ROM1}=\overline{CS}\&A_{17}\&\overline{A_{16}}$，

$\overline{CS}_{SRAM0}=\overline{CS}\&A_{17}\&A_{16}\&\overline{A_{15}}$，$\overline{CS}_{SRAM1}=\overline{CS}\&A_{17}\&\overline{A_{16}}\&A_{15}$，

$\overline{CS}_{SRAM2}=\overline{CS}\&A_{17}\&\overline{A_{16}}\&\overline{A_{15}}$，$\overline{CS}_{SRAM3}=\overline{CS}\&A_{17}\&A_{16}\&\overline{A_{15}}$，

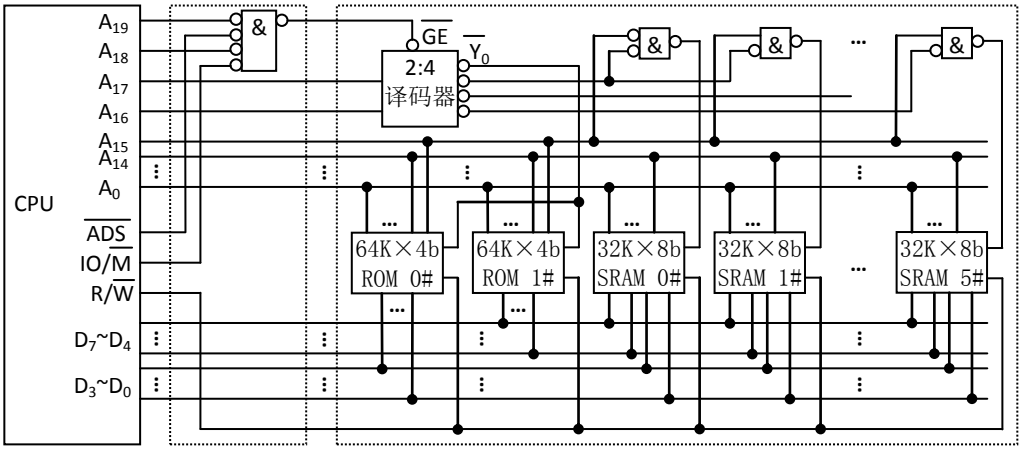
$\overline{CS}_{SRAM4}=\overline{CS}\&A_{17}\&A_{16}\&\overline{A_{15}}$，$\overline{CS}_{SRAM5}=\overline{CS}\&A_{17}\&A_{16}\&A_{15}$；

主存的信号线与内部各芯片引脚的连接如下图：



（3）由于主存空间通常与 CPU 存储空间的低端对应，故主存被选中的条件是：$\overline{ADS}$为低电平、IO/$\overline{M}$ 为低电平、A_{19} 及 A_{18} 为低电平；

CPU 与主存内部各芯片引脚的连接如下图：



13. 为什么说存储器的带宽比存储周期能更准确地反映性能？有哪些方法可以减小存储周期或平均存储周期，其原理是什么？在与存储周期或平均存储周期无关的情况下，又有哪些方法可以提高存储器的带宽，其原理是什么？

答：（1）存储周期指存储器连续进行两次访问的最短间隔时间，仅能反映存储器完成一次访问的性能，不能反映连续多次访问的性能，亦不能反映存储器数据引脚的性能；

而带宽指单位时间内存储器可以提供数据的速率，可以反映存储数据引脚及连续多次访问的性能，因此带宽比存储周期能更准确地反映存储器性能。

（2）扩展数据输出 DRAM(EDO DRAM)在常规 DRAM 中增加一个小容量 SRAM 缓冲区，减小了行地址相同的连续读操作的存储周期(行地址相同时直接从 SRAM 中读)；

同步 DRAM(SDRAM)基于时钟信号进行信号锁存及 I/O，缩短了异步方式的信号握手时间，猝发传输方式可缩短地址连续的多次访问的平均存储周期，同时可通过无须 CPU 等待的方法来隐藏 CPU 访存延迟；

多体交叉存储器的交叉工作方式使多个存储体轮流工作，减小了地址连续的多次访存的

平均存储周期。

（3）多体交叉存储器的并行工作方式使多个存储体同时工作，可在一个存储周期内访问多个存储字，提高了存储器带宽；

双端口存储器可同时进行地址不相同的两个操作，提高了存储器带宽。

14. 若某存储体有 8 位数据线、存储周期为 200ns，请问该存储体的带宽是多少？某多体存储器由 4 个这种存储体采用低位交叉编址构成，若多体存储器采用交叉工作方式，试问依次访问 64 个字时需要多少个存储周期？启动各存储体轮流工作的时钟频率是多少？

解：存储体带宽为：$8bit/(200\times10^{-9}s)=4\times10^{7}bit/s=40Mbps$；

设存储体的存储周期为 T_M，4 个存储体采用交叉工作方式时，存储体轮流工作的启动间隔为 $T_M/4$，依次访问 64 个字共需要 $T_M+(64-1)\times T_M/4=16.75T_M$；

启动各存储体轮流工作的时钟频率为：$1/(200ns/4)=2\times10^{7}Hz=20MHz$。

15. 为什么要求 Cache 支持按主存地址访问？Cache 与主存间以字块为信息交换单位可以减小 CPU 平均访问时间吗？若可以，请说明所需要的条件；若不可以，请说明理由。

解：（1）由于主存是 CPU 默认的存储器，故 CPU 按主存地址访问存储器；而 Cache 是为提高主存速度而设置的、位置介于 CPU 和主存之间，对 CPU 而言 Cache 是透明的，因此 CPU 按主存地址直接访问 Cache，即要求 Cache 支持按主存地址访问。

（2）可以。

设块大小＝n 个字、主存存储周期 $T_{MEM}=T_{地址}+T_{数据}$、从主存调入一个块的时间为 $T_{块}$，

存储系统靠程序访问具有局部性规律来减小 CPU 平均访问时间(T_A)，若 $T_{块}=nT_{MEM}$，字块一次调入还是多次调入不影响 T_A，则称 Cache 与主存间以字块为信息交换单位不影响 T_A；若 $T_{块}<nT_{MEM}$，则称 Cache 与主存间以字块为信息交换单位可以减小 T_A。

采用常规传送方式访问主存时 $T_{块}=nT_{MEM}=nT_{地址}+nT_{数据}$，采用猝发传送方式访问主存时 $T_{块}=T_{地址}+nT_{数据}<nT_{MEM}$，即 Cache 与主存间以字块为信息交换单位可以减小 T_A 的条件是：主存支持猝发传送访问方式。

16. 存储系统由 Cache 及主存组成，Cache 的存储周期为 40ns、调入块时采用请求字优先方式，主存的存储周期为 200ns，若 CPU 执行程序时共有 10000 次访存、Cache 有 50 次不命中。

（1）求 Cache 的命中率及 CPU 执行程序时的平均访问时间。

（2）相对于不设置 Cache，CPU 执行程序时的存储系统性能提高多少倍。

解：（1）Cache 的命中率为：$H=(10000-50)/10000=0.995$；

$T_A=HT_{命中}+(1-H)T_{不命中}$，由于块调入时采用请求字优先方式，故 $T_{不命中}=T_{MEM}=200ns$，

CPU 执行程序的平均访问时间为：$T_A=0.995*40ns+0.005*200ns=40.8ns$。

（2）存储系统性能加速比为：$S=T_M/T_A=200/40.8=4.902$ 倍，

即存储系统性能为主存的 4.902 倍。

17. 某计算机中，约定主存按字节编址，CPU 有 20 根地址引脚、8 根数据引脚，配置有 64KB 的 Cache，若 Cache 与主存采用全相联映像、块大小为 16B。

（1）画出 Cache、主存的地址组成格式，并计算出各组成部分的长度。

（2）每个块的块标记长度是多少？为什么？

（3）若 CPU 访存地址分别为 2D058H 和 2D078H，求 Cache 命中时对应的块标记是多少？

解：（1）由题意，主存单元长度为 8 位、主存地址长度为 20 位，

Cache 地址长度为 $\log_2(64KB/1B)=16$ 位，其中块内地址长度为 $\log_2(16B/1B)=4$ 位，

由于 Cache 与主存采用全相联映像，

则 Cache 地址组成格式如下：

块号(12b)	块内地址(4b)

主存地址组成格式如下：

块号(16b)	块内地址(4b)

（2）每个块的块标记长度为 16 位，因为全相联映像时某主存块可映射到 Cache 的任意块，故地址变换时需用主存地址中的块号与块标记比较。

（3）主存地址 2D058H＝$(0010\ 1101\ 0000\ 0101\ 1000)_2$ 中，块号(高 16 位)为 2D05H，则 Cache 命中时(任意块)对应的块标记为 2D05H；

主存地址 2D078H＝$(0010\ 1101\ 0000\ 0111\ 1000)_2$ 中，块号(高 16 位)为 2D08H，则 Cache 命中时(任意块)对应的块标记为 2D07H。

18. 某计算机的 CPU、主存编址单位、Cache 容量均与题 17 相同，若 Cache 与主存采用直接映像、块大小为 16B。回答与题 17 相同的三个问题。

解：（1）由题意，主存地址长度为 20 位、Cache 地址长度为 16 位，块内地址长度为 4 位，由于 Cache 与主存采用直接映像，

则 Cache 地址组成格式如下：

块号(12b)	块内地址(4b)

主存地址组成格式如下：

区号(4b)	区内块号(12b)	块内地址(4b)

（2）每个块的块标记长度为 4 位，因为直接映像时区内块号为 *i* 的主存块只可映射到 Cache 的第 *i* 块，故地址变换时只需用主存地址中的区号与块标记比较。

（3）主存地址 2D058H=$(0010\ 1101\ 0000\ 0101\ 1000)_2$ 中，区号(高 4 位)为 2H，则 Cache 命中时(第 D05H 块)对应的块标记为 2H；

主存地址 2D078H=$(0010\ 1101\ 0000\ 0111\ 1000)_2$ 中，区号(高 4 位)为 2H，则 Cache 命中时(第 D07H 块)对应的块标记为 2H。

19. 某计算机的 CPU、主存编址单位、Cache 容量均与题 17 相同，若 Cache 与主存采用 4 路组相联映像、块大小为 16B。回答与题 17 相同的三个问题。

解：（1）由题意，主存地址长度为 20 位、Cache 地址长度为 16 位，块内地址长度为 4 位，由于 Cache 与主存采用 4 路组相联映像，

则 Cache 地址组成格式如下：

组号(10b)	组内块号(2b)	块内地址(4b)

主存地址组成格式如下：

区号(6b)	区内块号(10b)	块内地址(4b)

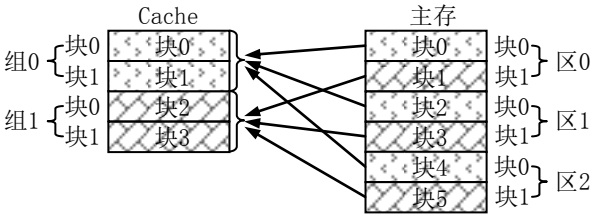
（2）每个块的块标记长度为 6 位，因为组相联映像时区内块号为 *i* 的主存块可映射到 Cache 的第 *i* 组的任意块，故地址变换时需用主存地址中的区号与组内每个块的块标记分别比较。

（3）主存地址 2D058H=$(0010\ 1101\ 0000\ 0101\ 1000)_2$ 中，区号(高 6 位)为$(0010\ 11)_2$，则 Cache 命中时(第 105H 组内任意块)对应的块标记为$(001011)_2$；

主存地址 2D078H=$(0010\ 1101\ 0000\ 0111\ 1000)_2$ 中，区号(高 6 位)为$(0010\ 11)_2$，则 Cache 命中时(第 107H 组内任意块)对应的块标记为$(001011)_2$。

20. 某 2 路组相联 Cache 的容量为 4 个块、块大小为 8 个字、采用 LRU 替换算法。假设 Cache 初态为全部空闲，CPU 先从地址 0000H 起升序连续访问 48 个字，再从地址 002FH 起降序连续访问 48 个字，求此时的 Cache 命中率。

解：由题意，CPU 访存地址范围为 0000H～002FH，共包含 48/8＝6 个主存块(块号为 0~5)，故解题时主存空间只需考虑该地址范围。2 路组相联地址映像效果如下：



由于 CPU 按连续地址访问主存，故 CPU 访存的块地址顺序为：0、1、2、3、4、5、5、4、3、2、1、0，每个块连续访问 8 次；

由于是 2 路组相联映像，故采用 LRU 替换算法时每个块的 LRU 位只需 $\log_2 2=1$ 位，若用 a-b-c 表示块状态，其中 a 为有效位，b 为 LRU 位，c 为块标记(区号)，则 CPU 按上述块地址顺序访存时的 Cache 各块状态变化如下表所示：

主存块地址流			0	1	2	3	4	5	5	4	3	2	1	0
块状态	组 0	块 0	1-0-0	1-0-0	1-1-0	1-1-0	1-0-4	1-0-4	1-0-4	1-0-4	1-0-4	1-1-4	1-1-4	1-0-0
		块 1	0-1-*	0-1-*	1-0-2	1-0-2	1-1-2	1-1-2	1-1-2	1-1-2	1-1-2	1-0-2	1-0-2	1-1-2
	组 1	块 0	0-1-*	1-0-1	1-0-1	1-1-1	1-1-1	1-0-5	1-0-5	1-0-5	1-1-5	1-1-5	1-0-1	1-0-1
		块 1	0-1-*	0-1-*	0-1-*	1-0-3	1-0-3	1-1-3	1-1-3	1-1-3	1-0-3	1-0-3	1-1-3	1-1-3
操作状态	第 1 次		调入	调入	调入	调入	替换	替换	命中	命中	命中	命中	替换	替换
	2~8 次		命中	命中	命中	命中	命中	命中	命中	命中	命中	命中	命中	命中

CPU 的 48*2＝96 次访存中，有 8 次不命中，则 Cache 命中率为(96－8)/96＝0.917。

21. 某计算机中最大主存容量为 2048 个块，实际主存容量配置为 1024 个块，Cache 容量为 256 个块，采用 4 路组相联映像、LRU 替换算法、写回法写策略。请计算目录表的最小容量。

解：主存块地址长度为 $\log_2 2048=11$ 位，与实际主存配置容量无关，

Cache 块地址长度为 $\log_2 256=8$ 位，

由于采用 4 路组相联映像，Cache 组内块号地址长度为 $\log_2 4=2$ 位，

Cache 块地址组成格式为：组号(6b)＋组内块号(2b)，

主存块地址组成格式为：区号(5b)＋区内块号(6b)，

故每个 Cache 块的块标记长度为 5b；

由于采用 LRU 替换算法，故每个 Cache 块的 LRU 位需 $\log_2 4=2$ 位；

由于采用写回法写策略，故每个 Cache 块需用 1 位脏位表示该块是否被改写过；

因此，目录表的最小容量为(5b＋2b＋1b)×256＝256B。

22. 为什么虚拟存储器中存在虚存-辅存地址变换？为什么虚拟存储器的管理表格通常都放在主存中？用虚拟地址查管理表有哪两个步骤？

解：（1）由于虚拟存储器是对正在执行的程序而言的，而所有程序均存储在辅存中，因此，虚存中所有信息均需从辅存中取得，故存在虚存-辅存地址变换。

（2）由于辅存速度太慢，主存-辅存间信息传送代价很大，为了尽可能不替换已调入主存的信息，虚存-主存间只能采用全相联映像方式；

又由于虚存空间巨大(可>>主存空间)，对这种全相联映像的虚存管理表格，硬件查表方式速度很快、但实现成本不可想象，故只能采用软件查表方式实现，即虚存-主存间管理采用以虚存地址为索引的索引表(非目录表)结构，这样一次查表即可获得结果、但管理表格更

大，因辅存速度太慢，故虚存管理表通常存放在主存中。

（3）由于虚存管理表采用索引表结构、管理表存放在主存中，因此，查管理表有①用管理表主存基址＋虚拟地址形成管理表项对应的主存索引地址，②用该主存索引地址访问主存获得管理表项内容这两个步骤。

23. 程序逻辑地址由 8 位段号及 16 位段内地址组成，主存按字节编址、最大容量为 512KB，若虚拟存储器采用段式管理，则虚存段表最多有多少行？只考虑映像的实现时，主存中段表每行至少有多少位？CPU 中快表的每行至少有多少位？

解：（1）由于虚存采用段式管理，故虚存段表最多有 2^8=256 行；

（2）由题意，主存地址长度为 $\log_2$(512KB/1B)＝19b，

由于只考虑映像的实现，故段表表项最少由装入位、段基址及段长组成，

即主存中段表每行至少有 1b＋19b＋16b＝36b；

（3）由于 CPU 中快表为目录表结构，故快表表项最少由有效位、段号、段基址及段长组成，即快表每行至少有 1b＋8b＋19b＋16b＝44b。

24. 程序逻辑地址及主存参数同题 22，若虚拟存储器采用页式管理、页大小为 4KB，则虚存页表最多有多少行？只考虑映像的实现时，页表每行至少有多少位？CPU 中快表的每行至少有多少位？

解：（1）由题意，页内地址长度为 $\log_2$(4KB/1B)＝12b，

程序逻辑页号长度为 8b＋16b－12b＝12b，故虚存页表最多有 2^{12}=4096 行；

（2）由题意，主存物理页号长度为 19b－12b＝7b，

由于只考虑映像的实现，故页表表项至少由装入位、物理页号组成，

即主存中页表每行至少有 1b＋7b＝8b＝1B；

（3）由于 CPU 中快表为目录表结构，故快表表项最少由有效位、逻辑页号及物理页号组成，即快表每行至少有 1b＋12b＋7b＝20b。

25. 虚拟存储器的地址映像及变换与 Cache 有哪些不同？

解：（1）虚拟存储器存在虚存-主存、虚存-辅存两种地址映像及变换，而 Cache 只存在主存-Cache 一种地址映像与变换；

（2）虚存-主存间信息交换粒度可为段或页，而 Cache-主存间信息交换粒度为块，块大小<<页大小或段大小；

（3）虚拟存储器只采用全相联映像方式，而 Cache 可采用直接映像、全相联映像或组相联映像方式；

（4）虚拟存储器的管理表存放在主存中、采用索引表结构，而 Cache 的管理表则存放在 Cache 中、采用目录表结构；

（5）虚拟存储器的查表及地址变换用软件方式实现，而 Cache 的查表及地址变换用硬件方式实现。

第 4 章

1. 解释概念或术语：机器指令、指令系统、指令格式、指令字长、扩展编码、堆栈、边界对齐、寻址方式、形式地址、规整性、兼容性、CISC、RISC。

答：略。

2. 为什么说指令系统是软件和硬件间的约定？约定有哪些内容？

答：（1）指令系统为所有机器指令的集合，包含所有机器指令的功能。

对计算机系统而言，为了保证应用需求与应用结果相一致，计算机软件(程序)必须是按照指令系统的约定(功能及格式)、根据应用需求而形成的，计算机硬件必须是按照指令系统的功能约定、根据指令格式进行设计而组成的。由此可见，指令系统是计算机软件和硬件间的约定。

（2）指令系统中，每条机器指令约定的信息需包括：操作类型、操作数类型、操作数长度、各操作数地址、各操作数的源/目标属性、下条指令地址等。

3. 指令字长、机器字长、存储字长三者间有何关系？为什么有些指令字中有空闲位置？

答：（1）由于 CPU 按地址访问存储器中数据，故机器字长必须是存储字长的倍数，为尽量节省数据所占存储空间，机器字长通常等于多个存储字长；

由于指令存放在存储器中，下条指令地址为存储器地址，故指令字长必须为存储字长的倍数，指令字长是否等于多个存储字长，取决于指令系统的风格及具体指令的约定。

（2）由于不同指令所需约定的信息不同，而所有指令字长必须是存储字长的倍数，因此，必然存在部分指令无需使用指令字长中的所有位，这就导致了有些指令字中有空闲位置。

4. 若某指令系统中共有 10 种性质的操作，使用频率分别为 0.25、0.20、0.15、0.10、0.08、0.08、0.05、0.04、0.03、0.02，请分别用定长编码法、霍夫曼编码法、同时有两种长度的霍夫曼扩展编码法进行编码，并计算三种方式编码的平均码长。

解：定长编码的位数为 $\log_2 10 = 4$ 位，霍夫曼编码及霍夫曼扩展编码根据霍夫曼树求得，编码如下表所示：

操作 I_i	频率 P_i	定长编码	霍夫曼编码	霍夫曼扩展编码
I_1	0.25	0000	00	000
I_2	0.20	0001	10	001
I_3	0.15	0010	010	010
I_4	0.10	0011	110	011
I_5	0.08	0100	0110	100
I_6	0.08	0101	1110	101
I_7	0.05	0110	01110	110
I_8	0.04	0111	01111	11100
I_9	0.03	1000	11110	11101
I_{10}	0.02	1001	11111	11110
平均码长($\sum I_i P_i$)		4	2.99	3.18
码长种类		1	4	2

注意：霍夫曼扩展编码亦可采用 2 位和 5 位，则平均码长＝0.6×2＋0.4×5＝3.2 位，可见表中答案较优。

5. 某指令系统中，指令字长为 16 位，指令操作码采用扩展编码法，有单地址指令、双地址指令两种指令。若每个地址码均为 6 位，且双地址指令为 A 条，则单地址指令最多有多少条？

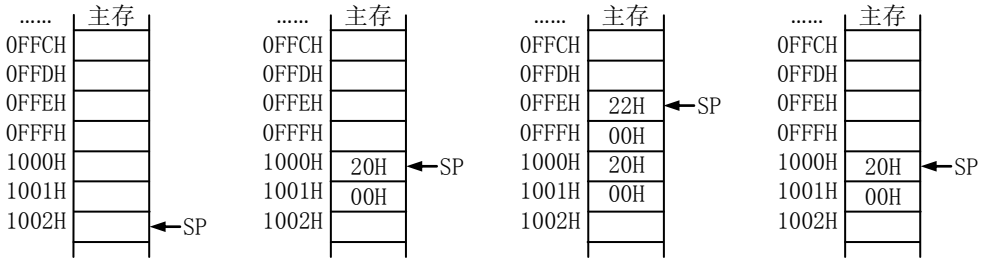
解：双地址指令操作码长为 16b－2×6b＝4b，由题意，空闲 2^4－A 种编码，

单地址指令操作码长为 16b－6b＝10b，可分为 4b 和 6b 两部分，

则单地址指令最多有$(2^4-A)\times 2^6$条。

6. 假设计算机约定主存按字节编址，数据在存储器中采用小端次序存放，CPU 中设置有 4 个通用寄存器(记为 R0~R3)，设置有寄存器 SP 指向存储器堆栈的栈顶，入栈操作时 SP 向递减方向移动。若(SP)=1002H、(R0)=0020H、(R1)=0022H、(R2)=0024H，连续进行将 R0 入栈、将 R1 入栈、出栈到 R2、将 R1 入栈、将 R2 入栈、出栈到 R0 操作后，请画出每一个操作后的 SP 及堆栈内部的数据变化，同时说明 R0~R2 的当前值。

解：堆栈操作时，堆栈内部、SP 及 R0~R2 的变化过程如图所示：

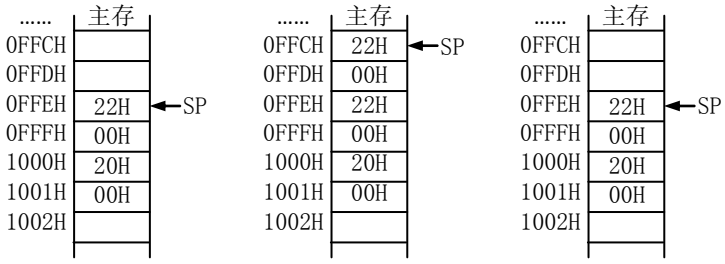


空堆栈　(SP)=1002H　(R0)=0020H　(R1)=0022H　(R2)=0024H

R0 入栈　(SP)=1000H　(R0)=0020H　(R1)=0022H　(R2)=0024H

R1 入栈　(SP)=0FFEH　(R0)=0020H　(R1)=0022H　(R2)=0024H

出栈到 R2　(SP)=1000H　(R0)=0020H　(R1)=0022H　(R2)=0022H



R1 入栈　(SP)=0FFEH　(R0)=0020H　(R1)=0022H　(R2)=0022H

R2 入栈　(SP)=0FFCH　(R0)=0020H　(R1)=0022H　(R2)=0022H

出栈到 R0　(SP)=0FFEH　(R0)=0022H　(R1)=0022H　(R2)=0022H

7. 指令寻址方式通常有哪些？顺序型指令和转移型指令的地址码内容有何不同？

答：（1）指令寻址方式通常有直接寻址、寄存器间接寻址、相对寻址、隐含寻址等；

（2）顺序型指令的功能是实现处理操作，故地址码中包含有操作数地址，下条指令地址为(PC)+“1”、与当前指令内容无关，可隐含表示，故地址码的内容仅为操作数地址码；转移型指令的功能仅是改变指令逻辑顺序，故地址码中不包含操作数地址码，而下条指令地址由当前指令形成、形成方法与当前指令内容有关，需显式表示，因此地址码内容仅为下条指令地址码。

8. 试比较间接寻址与寄存器间接寻址、基址寻址与变址寻址的不同点。

答：间接寻址与寄存器间接寻址不同点：虽然形式地址 A 均为存放存储器操作数地址的部件地址，但部件类型不同；操作数有效地址 EA 的形成性能不同，间接寻址的 EA＝[A]、需访存形成(性能差)，寄存器间接寻址的 EA＝(A)、无需访存形成(性能好)；寄存器间接寻址的形式地址 A 较短。

基址寻址与变址寻址的不同点：虽然两者均有 EA＝(REG)＋A，但所用 REG 不同，分别为从零编址的基址 REG 和变址 REG；有些机器的变址 REG 具有自动变址功能，而基址 REG 则没有，这是因为变址寻址主要用于数据块的规律操作，而基址寻址主要用于存储管理。

9. 设存储器按字节编址，相对寻址的转移指令字长为 2B，第二字节为相对偏移量，用补码表示。CPU 取指时每取出一个字节自动完成 PC←(PC)+1 操作。假设某相对寻址的转移指令的第一字节地址为 2000H，请分别写出欲转移到地址为 2008H 的指令和地址为 1FF8H 的指令时，该转移指令指令字的第二字节的内容是什么？

解：读取当前指令后(PC)＝2000H＋02H＝2002H，

（1）当前转移指令的下条指令 EA＝2008H 时，相对偏移量为 2008H－2002H＝+06H，

$[+06H]_{补}$＝06H，故当前指令字的第二字节内容为：06H；

（2）当前转移指令的下条指令 EA＝1FF8H 时，相对偏移量为 1FF8H－2002H＝-0AH，

$[-0AH]_{补}$＝F6H，故当前指令字的第二字节内容为：F6H。

10. 某机器字长为 16 位，存储器按字编址，指令字长同机器字长。指令格式如下：

5bit	3bit	8bit
操作码 OP	寻址方式位 MOD	形式地址 D

其中，MOD=000~100 分别表示立即寻址、直接寻址、间接寻址、变址寻址和相对寻址，变址寻址只使用唯一的变址寄存器(记为 RI)，形式地址 D 在 MOD＝000 及 100 时为补码表示、其余寻址方式时为无符号编码表示。

（1）该指令格式能定义多少种不同的操作？立即寻址的操作数范围是多少？

（2）写出各种寻址方式时，地址码对应的数据或地址表达式。

（3）写出 MOD=001、010、100 时，能访问的最大主存区为多少个机器字。

解：（1）由于操作码为 5 位，故可以定义 2^5＝32 种不同的操作；

由于立即寻址的 D 用 8 位补码表示，故立即寻址的操作数范围为 -128~+127。

（2）设 $D=[DZ]_{补}$，

则立即寻址的地址码表示的是：操作数＝DZ；

直接寻址的地址码表示的是：操作数 EA＝D；

间接寻址的地址码表示的是：操作数 EA＝[D]；

变址寻址的地址码表示的是：操作数 EA=(RI)+D；

相对寻址的地址码表示的是：下条指令 EA=(PC)+DZ。

（3）MOD＝001(直接寻址)时，操作数 EA 长度＝D 的长度＝8bit，

可访问的主存区为 2^8＝256 个机器字；

MOD＝010(间接寻址)时，操作数 EA 长度＝存储字长＝机器字长＝16bit，

可访问的主存区为 2^{16}＝64K 个机器字；

MOD＝100(相对寻址)时，下条指令 EA 长度＝max{PC 的长度，D 的长度}＝max{16bit，8bit}＝16bit，可访问的主存区为 2^{16}＝64K 个机器字。

11. 某计算机约定数值数据采用定点表示，有符号数采用原码编码方式，主存按字节编址，机器字长为 8 位。CPU 中有 4 个通用寄存器(记为 R0～R3)可被指令使用，CPU 处理的定

点数长度只有 8 位这一种，CPU 取指令时每取出一个字节自动完成 PC←(PC)+1。指令系统的部分指令格式如图 4.20 所示。请按指令格式要求编写求 Y＝A[19]+…+ A[0]的程序，其中 A、Y 分别为主存中数组和变量。

解：假设 A 的地址为 10H、Y 的地址为 24H，则程序如下图所示：

地址	程序	对应的C语言程序
00H	000000**	BYTE R0=0;
01H	00000000	
02H	000001**	BYTE R1=14H; //20
03H	00010100	
04H	000010**	BYTE R2=23H; //&A[19]
05H	00100011	
06H	000011**	BYTE R3=24H; //&Y
07H	00100100	do
08H	01010010	{ R0=R0+*R2;
09H	011110**	R2--;
0AH	011101**	R1--;
0BH	1000****	} while (R1≠0);
0CH	00001000	
0DH	00111100	*R3=R0;
	……	
10H	**	//A[0]
	……	
23H	**	//A[19]
24H	**	//Y

12. 试简单比较 CISC 机及 RISC 机的优缺点。

答：程序执行时间 T_{CPU}＝I_N×CPI×T_C，CISC 机侧重增强指令功能，减少程序中指令数 I_N 以提高性能，而 RISC 机侧重简化指令功能、提高指令执行速度 CPI 以提高性能。

CISC 机采用变长指令字结构，因而指令系统的可扩展性及兼容性较好；由于支持较多种寻址方式，大多数指令支持存储器操作数，因而程序的代码效率较高，只需设置少量寄存器，但指令执行时间较长，编译程序的设计相对复杂；由于指令系统较为复杂，因而指令的执行控制复杂，不利于使用 VLSI 技术。

RISC 机采用定长指令字结构，因而指令中操作数较多，有利于扩充指令功能，但指令系统的可扩展性及兼容性较差；由于支持的寻址方式种类较少，除 LOAD/STORE 指令外，其余指令的操作数均为寄存器类型，因而指令格式规整，指令译码及执行速度很快，但需要设置大量通用寄存器，又导致编译程序简单；由于指令系统简单，指令的执行控制简单，有利于使用 VLSI 技术，有利于提高主频。

第 5 章

1. 解释概念或术语：ALU、CU、BIU、MAR、MDR、指令周期、CPU 周期、微操作、微操作步、数据通路、机器周期、节拍周期、节拍脉冲、微命令、微指令、微程序、CM、毫微程序、动态微程序、并行性、线性流水线、动态流水线、超级流水线、超标量流水线、吞吐率、加速比、效率、结构相关、数据相关、控制相关、RAW 相关。

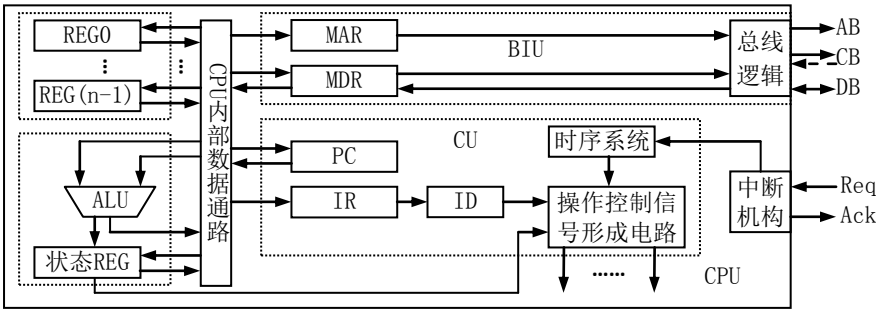
答：略

2. CPU 有哪些功能？说明实现这些功能各需要哪些部件，并画出 CPU 基本结构图。

答：CPU 的基本功能有指令控制、操作控制、时间控制、数据加工和中断处理 5 大功能，除数据加工外，其余功能均由控制器实现。

指令控制功能的实现，需要程序计数器 PC、指令寄存器 IR、及指令译码器 ID 支持；操作控制和时间控制功能的实现，需要时序系统、微操作控制信号形成部件支持；数据加工功能的实现，需要算术逻辑运算部件 ALU、状态寄存器、及数据寄存器支持；中断处理功能的实现，需要中断处理机构的支持。

CPU 基本结构图如下图所示：



3. CPU 内部有哪些基本操作？这些基本操作各包含哪些微操作？

答：CPU 内部的基本操作有寄存器间数据传送、存储器读、存储器写、算逻运算 4 种。

（1）寄存器间数据传送为原子操作，微操作序列为：$R_{源}$→$R_{目标}$。

（2）存储器读的微操作序列为：①1→Read； ②M(MAR)→MDR。

（3）存储器写的微操作序列为：①1→Write；②MDR→M(MAR)。

（4）算逻运算的微操作序列为：①$R_{源1}$→ALU；②$R_{源2}$→ALU；
③OP→ALU； ④ALU→$R_{结果}$。

4. 若将图 5.8 中 12H 主存单元的内容改为 01010011B，写出该指令执行过程的微操作序列。

解：执行 12H 主存单元中指令的微操作序列为

① PC→MAR
② 1→Read，(PC)+1→PC ；1→Read 包含(MAR)→AB、Read→CB 操作
③ M(MAR)→MDR
④ MDR→IR ；ID 译码后发现指令操作为 R0←(R0)+[(R3)]
⑤ R3→MAR
⑥ 1→Read
⑦ M(MAR)→MDR
⑧ MDR→ALU

⑨ R0→ALU，ADD→ALU

⑩ ALU→R0

5. CPU 结构如图 5.60 所示，a～d 为 4 个寄存器，各部分之间的连线为数据通路，箭头表示信息传送方向。

（1）说明图中 a～d 这 4 个寄存器的名称。

（2）简述从开始取指令到产生控制信号的数据流动的过程。

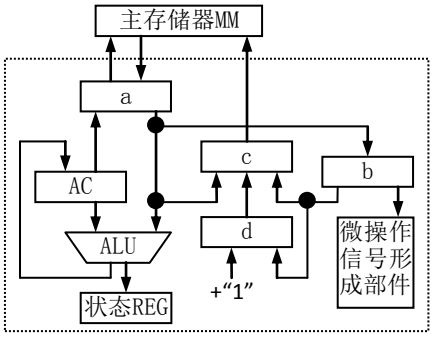


图5.60 基于累加器CPU例

解：（1）a 为存储器数据寄存器(MDR)，b 为指令寄存器(IR)，c 为存储器地址寄存器(MAR)，d 为程序计数器(PC)。

（2）① PC→MAR；

② MAR→MM，Read→MM，(PC)+1→PC；

③ MM→MDR；

④ MDR→IR；

⑤ IR→微操作信号形成部件。

6. 为什么说数据通路会影响指令执行的性能？单总线数据通路对基于寄存器的 CPU 中运算器的组织有何要求？为什么？基本算逻运算的微操作步序列是什么？

答：由于数据通路的个数决定了每个微操作步中传送操作微操作的最大数量，而具体指令执行过程中的数据传送微操作的数量是固定的，因此，数据通路会影响指令执行过程的微操作步序列的步数；又由于指令执行时间＝Σ微操作步执行时间，因此数据通路会影响了指令执行的性能。

单总线数据通路要求运算器的 2 个入端、1 个出端不同时使用总线通路，即要求运算器设置一个入端暂存器(常记为 Y)、一个出端寄存器(常记为 Z)，或两个入端暂存器。这是因为 ALU 是组合逻辑部件，只能加工数据、不能存储数据。

基本算逻运算的微操作步序列为：

① $R_{源1}$→Y

② $R_{源2}$→ALU，OP→ALU，ALU→Z

7. 单总线数据通路的 CPU 结构如图 5.11(b)所示，PC 具有计数功能。写出下列指令的微操作步序列。

（1）单字长指令[(R2)]←(R1)+(R0)，目标数操作为寄存器间接寻址；

（2）单字长指令 R1←(R1)+[(R0)+DISP]，DISP 为变址寻址的偏移量；

（3）双字长指令 R1←(R1)+[(R0)+DISP]；

（4）单字长指令[(R1)+DISP]←[(R1)+DISP]+(R0)。

解：取指令的微操作步序列为

t1: PC→MAR，　1→Read

t2: (PC)+1→PC，M(MAR)→MDR

t3: MDR→IR

（1）单字长指令[(R2)]←(R1)+(R0)的微操作步序列为

t1～t3: 同取指令微操作步序列

t4: R0→Y

t5: R1→ALU，ADD→ALU，ALU→Z

t6: R2→MAR，1→Write

t7: Z→MDR， MDR→M(MAR)

t8: 1→End

（2）单字长指令 R1←(R1)+[(R0)+DISP]的微操作步序列为

t1～t3: 同取指令微操作步序列

t4: R0→Y

t5: IR(DISP)→ALU, ADD→ALU，ALU→Z

t6: Z→MAR，1→Read

t7: R1→Y， M(MAR)→MDR

t8: MDR→ALU, ADD→ALU，ALU→Z

t9: Z→R1，1→End

（3）双字长指令 R1←(R1)+[(R0)+DISP]的微操作步序列为

t1～t3: 同取指令微操作步序列

t4: PC→MAR， 1→Read

t5: (PC)+1→PC，M(MAR)→MDR

t6: MDR→Y ；设指令第二字内容为 DISP

t7: R0→ALU，ADD→ALU，ALU→Z

t8: Z→MAR，1→Read

t9: R1→Y， M(MAR)→MDR

t10: MDR→ALU，ADD→ALU，ALU→Z

t11: Z→R1，1→End

（4）单字长指令[(R1)+DISP]←[(R1)+DISP]+(R0)的微操作步序列为

t1～t3: 同取指令微操作步序列

t4: R1→Y

t5: IR(DISP)→ALU, ADD→ALU，ALU→Z

t6: Z→R2，Z→MAR，1→Read ；设用 R2 暂存目标操作数地址

t7: R0→Y， M(MAR)→MDR

t8: MDR→ALU, ADD→ALU，ALU→Z

t9: R2→MAR，1→Write

t10: Z→MDR， MDR→M(MAR)

t11: 1→End

8. 单总线数据通路的 CPU 结构如图 5.11(b)所示，PC 具有计数功能，CPU 中还设置有堆栈指针寄存器 SP(图中未标出)。过程调用指令 CALL [(R0)]为寄存器间接寻址的单字长指令，约定(R0)为所调用过程的入口地址、过程返回地址保存到主存堆栈中，堆栈栈顶由 SP 指向；返回指令 RET 为无显式操作数的单字长指令，返回地址存放在 SP 指向的主存堆栈顶部。请写出这两条指令的微操作步序列。

解：CALL [(R0)]指令的微操作步序列为

t1: PC→MAR， 1→Read

t2: (PC)+1→PC，M(MAR)→MDR

t3: MDR→IR ；ID 译码后发现为 CALL 指令

t4: (SP)-1→SP ；假设存储字长＝机器字长，否则为“-*x*”

；假设入栈操作时 SP 向递减方向移动

t5: SP→MAR，1→Write

t6: PC→MDR，MDR→M(MAR) ；(PC)已进行过“+1”，为下条指令地址

t7: R0→PC，1→End

RET 指令的微操作步序列为

t1: PC→MAR，　1→Read

t2: (PC)+1→PC，M(MAR)→MDR

t3: MDR→IR　　；ID 译码后发现为 RET 指令

t4: SP→MAR，　1→Read

t5: (SP)+1→SP，M(MAR)→MDR

t6: MDR→PC，1→End

9. 某双总线 CPU 结构如图 5.61 所示，ALU 有 F＝A＋B 及 F＝A 两个功能，指令系统如图 5.5 所示，请写出指令系统中 MOV、ST、ADD 及 JNZ 指令的微操作步序列。

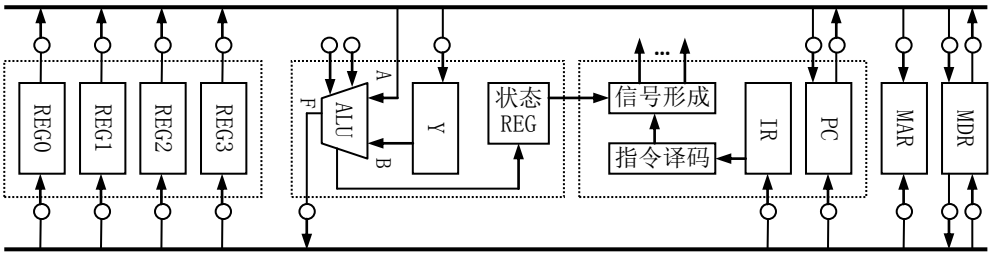


图 5.61 双总线 CPU 结构例

解：（1）MOV 指令的微操作步序列为

t1: PC→MAR，　1→Read

t2: (PC)+1→PC，M(MAR)→MDR

t3: MDR→IR

立即寻址方式时

t4: PC→MAR，　1→Read

t5: (PC)+1→PC，M(MAR)→MDR

t6: MDR→RD，1→End

寄存器寻址方式时

t4: RS→ALU，(F＝A)→ALU，ALU→RD，1→End

（2）ST 指令的微操作步序列为

t1～t3: 同 MOV 指令取指令微操作步序列

t4: RD→MAR，1→Write

t5: RS→MDR，MDR→M(MAR)

t6: 1→End

（3）ADD 指令的微操作步序列为

t1～t3: 同 MOV 指令取指令微操作步序列

寄存器寻址方式时

t4: RD→Y

t5: RS→ALU，(F＝A＋B)→ALU，ALU→RD，1→End

寄存器间接寻址方式时

t4: RS→MAR，1→Read

t5: RD→Y，　M(MAR)→MDR

t6: MDR→ALU，(F＝A＋B)→ALU，ALU→RD，1→End

（4）JNZ 指令的微操作步序列为

t1～t3: 同 MOV 指令取指令微操作步序列

Z≠0(即运算结果为零)时

t4： 1→End

Z＝0(即运算结果不为零)时

t4： PC→Y

t5： IR(DISP)→ALU，(F＝A＋B)→ALU，ALU→PC，1→End

10. 联合控制方式实现同步控制与异步控制间转换的思路是什么？需要哪些微操作控制或状态信号？实现相互转换的原理是什么？

答：实现思路是延长节拍周期，即节拍周期长度为异步方式长度对应的节拍周期数。

通常需要 1 个微操作控制信号表示是否已转入异步控制方式，需要 1 个状态信号表示异步控制方式是否已结束。以访存操作为例，分别为 WMFC 和 MFC。

实现原理是异步控制方式时封锁节拍周期信号发生器的 CP，即 CP＝CLK&CTL、CTL＝$\overline{\text{WMFC}}$＋WMFC&MFC。即同步方式时(WMFC＝0)，信号发生器 CP＝CLK；转入异步方式时(存储器使 MFC＝0、控制器使 WMFC＝1)，信号发生器 CP＝0 被封锁；异步方式结束时(存储器使 MFC＝1、控制器使 WMFC＝1)，信号发生器 CP＝CLK 封锁被解除，恢复到同步方式(下个节拍周期使 WMFC＝0)。

11. 某 CPU 主频为 200MHz，每个机器周期平均包含 4 个节拍周期，每个指令周期平均包含 3 个机器周期(其中有 2 个机器周期需要访存)。

（1）若每次访存时，机器周期中不需要插入等待周期，求该机的平均指令周期。

（2）若每次访存时，机器周期中需要插入 2 个等待周期，求该机的平均指令周期。

解：由于 CPU 主频为 200MHz，故节拍周期＝时钟周期＝1/(200×10^6)＝5ns。

（1）机器周期＝4*5ns＝20ns，平均指令周期＝3*20ns＝60ns。

（2）访存的机器周期＝(4＋2)*5ns＝30ns，不访存的机器周期＝20ns，

平均指令周期＝2*30ns＋1*20ns＝80ns。

12. 若 CPU 结构如图 5.11(b)所示，分别写出 4 种 CPU 基本操作的微操作命令序列。

解：假设 CPU 内部与外部采用联合控制方式，微操作控制信号为 WMFC；

（1）寄存器间数据传送(Rx→Ry)的微操作命令序列为：Rx_{out}、Ry_{in}；

（2）存储器读的微操作命令序列为：①Read， ②WMFC；

（3）存储器写的微操作命令序列为：①Write，②WMFC；

（4）算逻运算(R1+R2)的微操作命令序列为：①$R1_{out}$、Y_{in}，②$R2_{out}$、ADD、Z_{in}。

13. 写出题 7 的微操作步序列对应的微操作命令序列。

解：取指令的微操作命令序列为

t1： PC_{out}、MAR_{in}，Read

t2： PC_{+1}，WMFC

t3： MDR_{out}、IR_{in}

（1）单字长指令[(R2)]←(R1)+(R0)的微操作命令序列为

t1～t3：同取指令微操作命令序列

t4： $R0_{out}$、Y_{in}

t5： $R1_{out}$、ADD、Z_{in}

t6： $R2_{out}$、MAR_{in}，Write

t7： Z_{out}、MDR_{in}，WMFC

t8： End

（2）单字长指令 R1←(R1)+[(R0)+DISP]的微操作命令序列为

t1～t3：同取指令微操作命令序列

t4： $R0_{out}$、Y_{in}

t5： SE_{out}、ADD、Z_{in}

t6： Z_{out}、MAR_{in}，Read

t7： $R1_{out}$、Y_{in}，WMFC

t8： MDR_{out}、ADD、Z_{in}

t9： Z_{out}、$R1_{in}$，End

（3）双字长指令 R1←(R1)+[(R0)+DISP]的微操作命令序列为

t1～t3：同取指令微操作命令序列

t4： PC_{out}、MAR_{in}，Read

t5： PC_{+1}，WMFC

t6： MDR_{out}、Y_{in}

t7： $R0_{out}$、ADD、Z_{in}

t8： Z_{out}、MAR_{in}，Read

t9： $R1_{out}$、Y_{in}，WMFC

t10：MDR_{out}、ADD、Z_{in}

t11：Z_{out}、$R1_{in}$，End

（4）单字长指令[(R1)+DISP]←[(R1)+DISP]+(R0)的微操作命令序列为

t1～t3：同取指令微操作命令序列

t4： $R1_{out}$、Y_{in}

t5： SE_{out}、ADD、Z_{in}

t6： Z_{out}、$R2_{in}$、MAR_{in}，Read

t7： $R0_{out}$、Y_{in}，WMFC

t8： MDR_{out}、ADD、Z_{in}

t9： $R2_{out}$、MAR_{in}，Write

t10：Z_{out}、MDR_{in}，WMFC

t11：End

14. 写出题 8 的微操作步序列对应的微操作命令序列。

解：CALL [(R0)]指令的微操作命令序列为

t1： PC_{out}、MAR_{in}，Read

t2： PC_{+1}，WMFC

t3： MDR_{out}、IR_{in}

t4： SP_{-1}

t5： SP_{out}、MAR_{in}，Write

t6： PC_{out}、MDR_{in}，WMFC

t7： $R0_{out}$、PC_{in}，End

RET 指令的微操作命令序列为

t1～t3：同取指令微操作命令序列

t4： SP_{out}、MAR_{in}，Read

t5： SP_{+1}，WMFC

t6： MDR_{out}、PC_{in}，End

15. 某单总线通路的、基于累加器的 CPU 如图 5.62 所示，控制器采用硬布线控制方式，CPU 工作流程暂不考虑中断与 DMA 处理，若信号时序采用联合控制方式，同步控制方式的时序中暂不考虑节拍脉冲。该 CPU 的指令系统由 5 条单字长指令组成。请设计组合逻辑控制器

的微操作控制信号形成电路。

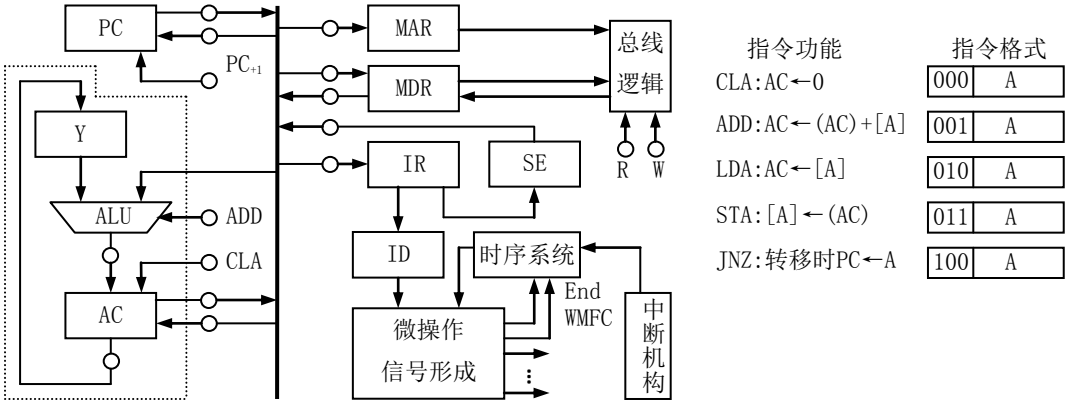


图 5.62 CPU 结构及指令系统例

解：该控制器的微操作控制信号形成电路设计的过程如下：

第一步，列出所有的微操作命令序列。

由于不考虑中断及 DMA 处理，5 条指令格式共有 6 个微操作命令序列，其中取指周期的微操作命令序列是通用的。

取指令阶段的微操作命令序列如下：

t1：PC_{out}、MAR_{in}，Read

t2：PC_{+1}，WMFC

t3：MDR_{out}、IR_{in}

CLA 指令执行阶段的微操作命令序列如下：

t4：CLA，End

ADD 指令执行阶段的微操作命令序列如下：

t4：SE_{out}、MAR_{in}，Read

t5：Y_{in}，WMFC

t6：MDR_{out}、ADD、AC_{ALUin}，End

LDA 指令执行阶段的微操作命令序列如下：

t4：SE_{out}、MAR_{in}，Read

t5：WMFC

t6：MDR_{out}、AC_{in}，End

STA 指令执行阶段的微操作命令序列如下：

t4：SE_{out}、MAR_{in}，Write

t5：AC_{out}、MDR_{in}，WMFC

t6：End

JNZ 指令执行阶段的微操作命令序列如下：

Z＝0 时　　Z≠0 时

t4：SE_{out}、PC_{in}，End　　t4：End

第二步，确定时序系统相关参数。

由上述微操作命令序列可以看出，序列的每一行为一个节拍周期，执行阶段的节拍周期数有 3 个和 1 个两种。因题目无要求，时序系统的参数确定为：指令周期由取指(IF)、执行(EX) 2 个机器周期组成，每个机器周期由 3 个节拍周期组成。

第三步，形成所有微操作命令的使用时间表及有效逻辑表达式。

首先微调微操作命令序列，将 CLA、JNZ 指令中的 End 放在 t6 步，其余不变。

同时用机器周期 IF/DF/EX 和节拍周期 $T_0/T_1/T_2$代替序列中的 t1、t2、…。

满足指令系统需求的所有微操作命令的信号-时序二维表如下，其中 IF 的 $T_0/T_1/T_2$ 等价于 t1、t2、t3，EX 的 $T_0/T_1/T_2$ 等价于 t4、t5、t6，“|”表示“逻辑或”。

	IF			EX		
	T_0	T_1	T_2	T_0	T_1	T_2
PC_{out}	ALL					
PC_{in}				JNZ($\overline{Z}$)		
PC_{+1}		ALL				
IR_{in}			ALL			
SE_{out}				ADD|LDA|STA| JNZ($\overline{Z}$)		
CLA				CLA		
AC_{out}					STA	
AC_{in}						LDA
AC_{ALUin}						ADD
Y_{in}					ADD	
ADD						ADD
MAR_{in}	ALL			ADD|LDA|STA		
MDR_{out}			ALL			ADD|LDA
MDR_{in}					STA	
Read	ALL			ADD|LDA		
Write				STA		
WMFC		ALL			ADD|LDA|STA	
End						ALL

根据上表，所有微操作命令的有效逻辑表达式如下：

PC_{out}＝IF&T_0

PC_{in} ＝JNZ& $\overline{Z}$ &EX&T_0

…

SE_{out}＝(ADD＋LDA＋STA＋JNZ& $\overline{Z}$)&EX&T_0

…

MAR_{in}＝IF&T_0＋(ADD＋LDA＋STA)&EX&T_0

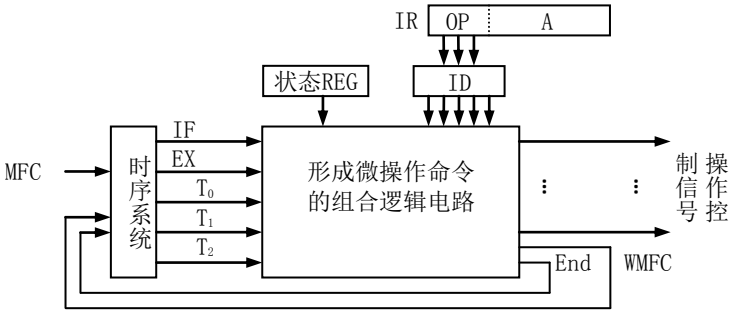
…

End ＝EX&T_2

第四步，画出微操作控制信号形成电路及与相关部件的连接图。

可以简单地用 18 个组合逻辑电路实现上表中 18 个微操作命令的有效逻辑达式。至此，微操作控制信号形成电路的所有微操作控制信号已经形成。

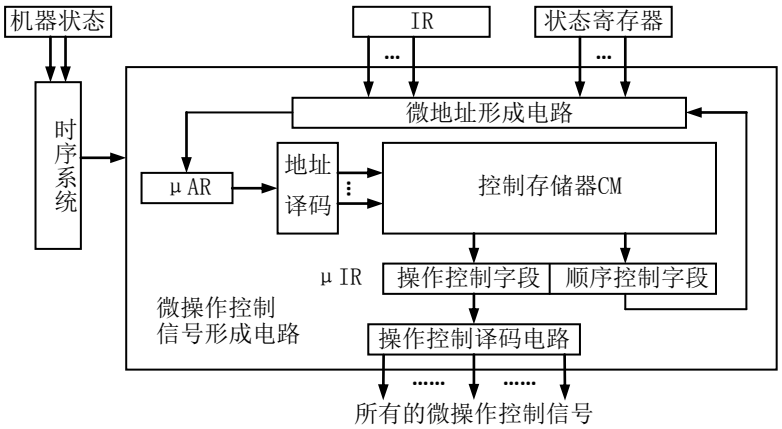
微操作控制信号形成电路与相关部件的连接如下图所示：



16. 简述微程序控制的基本思想。相对于组合逻辑控制器，微程序控制器的微操作控制信号形成电路由哪些基本部件组成？

答：微程序控制的基本思想是：将 CPU 工作流程对应的每个微操作命令序列编写成微程序(微程序由若干微指令组成、每条微指令对应一个或几个微操作命令)，所有的微程序存放一个只读存储器中；控制器自动按照微程序的逻辑顺序，逐条取出微指令并执行，以实现 CPU 工作流程的控制。

微程序控制器的微操作控制信号形成电路由微程序存储部件（CM）及微指令部件（μAR、μIR、微操作控制电路、微地址形成电路）组成。具体结构如下图所示：



17. 假设某机器有 45 种机器指令平均由 6 条微指令组成，其余 30 种机器指令平均由 10 条微指令组成，其中有 3 条取指微指令是所有指令公用的。已知微指令长度为 64 位，请估算控制存储器的最小容量及微指令地址位数。

解：所有机器指令的微指令条数共有：45×(6-3)+30×(10-3)+3＝348 条；

控制存储器的最小容量为：348×64bit＝348×64/8＝2784B；

由于 256＜348＜512，故微指令地址位数为 $\log_2 512$＝9 位。

18. 简述水平型微指令格式和垂直型微指令格式的特点。

答：水平型微指令格式能够同时定义并执行多个微命令(≤全部微命令数)，微指令的微操作能力强、灵活性强、效率高，机器指令执行速度快，代码效率较低(少数有效)；

垂直型微指令格式能够同时定义并执行一个或几个微命令(有限几个)，微指令格式由微操作码及微地址码组成，代码效率高，但微指令的微操作能力、灵活性、效率均较弱。

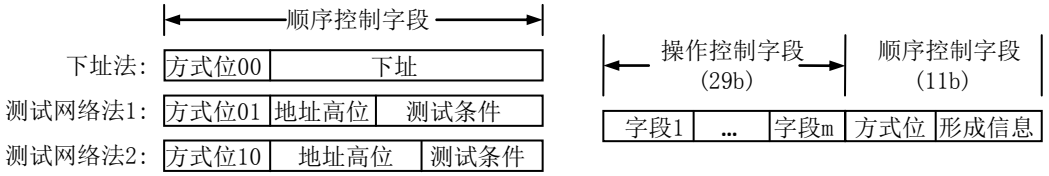
19. 某微程序控制器中，CM 容量为 512×40 位，微指令采用水平型格式，顺序控制字段采用下址法+测试网络法(断定法)，测试网络有 2 种外部测试条件。

（1）请设计该微程序控制器的微指令格式。

（2）该微程序控制器中最少有多少个微命令。

解：CM 的地址为 $\log_2 512$＝9 位；

（1）微指令格式顺序控制字段中，寻址方式位为 $\log_2(1+2)$＝2 位，下址为 9 位，则顺序控制字段为 2+9＝11 位，操作控制字段为 40－11＝29 位，顺序控制字段及微指令格式如下图所示：



（2）该微程序控制器中最少有 29 个微命令，最多有 2^{29} 个微命令。

20. 某微程序控制器中，微指令采用水平型格式、长度为 26 位，操作控制字段采用字段直接编码法，共 5 组，每组共有 5 个、8 个、15 个、27 个、3 个微命令；顺序控制字段采用下址法+测试网络法(断定法)，测试网络有 2 种外部测试条件。

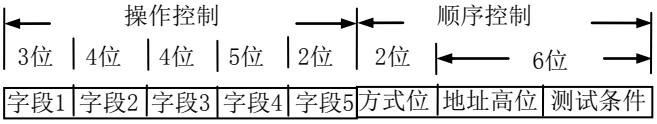
（1）请设计该微程序控制器的微指令格式。

（2）控制存储器 CM 的容量最大应为多少？

解：(1) 操作控制字段中，由于每组分别有 5 个、8 个、15 个、27 个、3 个微命令，而每组需各留出一种编码表示微命令全部无效，故每组字段编码位数分别为 3 位、4 位、4 位、5 位、2 位。

顺序控制字段中，由于微指令地址有 3 种形成方法，故寻址方式为 $\log_2(1+2)=2$ 位，顺序控制字段总长为 26－3－4－4－5－2＝8 位。

该微程序控制器的微指令格式为：



（2）由于顺序控制字段总长为 8 位，其中方式位为 2 位，则下址法中的微指令地址长度为 8－2＝6 位，即 CM 地址为 6 位，CM 容量最大应为 $2^6\times 26\text{bit}=208\text{B}$。

21. 将指令执行过程分为取指 IF、译码 ID、取数 OF、执行 EX、写结果 WB 共 5 个阶段，各阶段的操作时长分别为 10ns、5ns、10ns、8ns、10ns。若将这 5 个阶段组织成线性流水线，则

（1）流水线的拍长应为多少？

（2）采用串行方式执行 10000 条指令时，共需要多少时间？

（3）若用流水方式执行同样数量指令，流水线的吞吐率、加速比、效率各是多少？

解：（1）流水线拍长＝max{10ns，5ns，10ns，8ns，10ns }＝10ns；

（2）若采用串行方式，每条指令执行时间为 10ns＋5ns＋10ns＋8ns＋10ns＝43ns，
执行 10000 条指令共需要 10000×43ns＝0.43ms；

（3）$T_{流水}=5\Delta t+(10000-1)\Delta t=10004\times 10\text{ns}$，
吞吐率 $T_p=10000/T_{流水}=10000/(10004\times 10\text{ns})\approx 10^8$，
加速比 $S=T_{串行}/T_{流水}=0.43\text{ms}/(10004\times 10\text{ns})\approx 4.3$，
效率 $E=(10000\times 5\times 10\text{ns})/[5\times(10004\times 10\text{ns})]\approx 1$。

22. 简述解决流水线结构相关中访存冲突的方法。

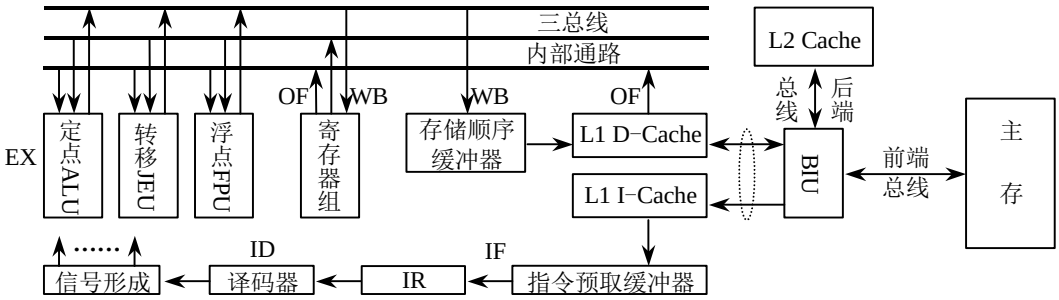
解：指令流水线通常包含有取指(IF)、译码(ID)、取操作数(OF)、执行(EX)、写结果(WB)这几个段，解决访存冲突实际上是解决 IF、OF 及 WB 段的冲突。通常有 4 种方法，其中第 4 种方法是必须的，其余的都可以不采用。

⑴增设一个存储器，两个存储器分别存放指令和数据。如现代计算机中 L1 级 Cache 采用哈佛结构，大大降低了 IF 段与 OF/WB 段的访存冲突概率。

⑵增设数据存储缓冲器(记为 MOB)，暂存 WB 段需写数据，在 OF 段不访存时再写回存储器。大大降低了 OF 段与 WB 段的访存冲突概率(MOB 满时才可能产生冲突)。

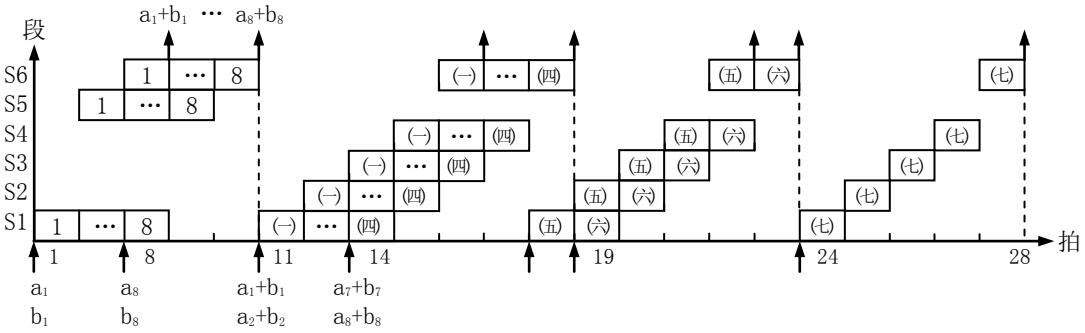
⑶增设指令预取缓冲器，采用指令预取技术。由于指令缓冲器有空闲位置时就预取，降低了指令存储器(如 L1$)访问通用存储器(如 MEM)的访存优先级。

⑷各个冲突段串行访问存储器，彻底解决访存冲突，通常使 IF 段的访存优先级低于 OF/WB 段。



23. 若将例 5.9 的计算要求改为$\prod_{i=1}^{8}(a_i+b_i)$，请计算实现该运算时流水线的吞吐率、加速比和效率。

解：适合流水线的解题算法是先计算所有 a_i+b_i、再用折叠法实现累乘求积。由于是静态流水线，功能切换时必须先排空流水线。流水线完成题目要求的时-空图如下图所示：



算法共需 15 条指令，流水方式运算时间 $T_{流水}=28\Delta t$，

串行方式运算时间 $T_{串行}=8\times3\Delta t+7\times5\Delta t=59\Delta t$；

吞吐率 $T_p=15/(28\Delta t)$；

加速比 $S_p=T_{串行}/T_{流水}=(59\Delta t)/(28\Delta t)=2$；

效率 $E=(59\Delta t)/(6\times28\Delta t)=1/3$。

24. 某 5 段线性、动态流水线如图 5.63 所示，转移型指令在 EX 段形成转移目标地址、在 WB 段写入 PC。数据相关采用重定向法，控制相关可以采用冻结法+重定向法、或者静态分支预测法。采用静态分支预测法处理时，总是预测转移不发生，预测错误时回头准备工作在 WB 段完成。分别计算当流水线的控制相关处理采用冻结法+重定向法和静态分支预测法时，下列程序段的执行时间。

```
        R0←0
        R2←1
        R3←100
LOOP1:  R1←(R2)*(R2)
        R0←(R0)+(R1)
        R2←(R2)+1
        R3←(R3)-1
        JNZ  LOOP1          ；状态寄存器(Z)=0 时(即前驱算术运算结果不位零)转移
        R0←(R0)/100
```

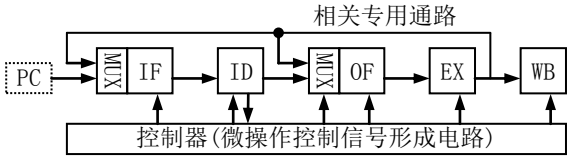


图5.63　流水线例

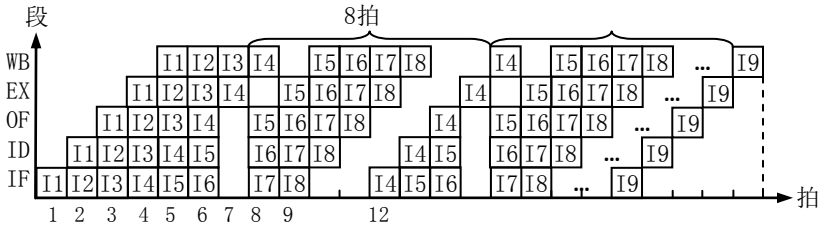
解：上述程序中，由于指令 R1←(R2)*(R2)与指令 R0←(R0)+(R1)存在 RAW 相关，采用重定向法解决时流水线将停顿 1 拍，如右图所示：

拍	1	2	3	4	5	6	7	8
I1	IF	ID	OF	EX	WB			
I2	IF	ID		OF	EX	WB		

（1）控制相关处理采用冻结法+重定向法解决时，JNZ 指令导致流水线停顿 3 拍：

指令 \ 拍	1	2	3	4	5	6
I1：JNZ LOOP1	IF	ID	OF	EX	WB	
Ix：???					IF	ID

上述程序执行时流水线时-空图如下：

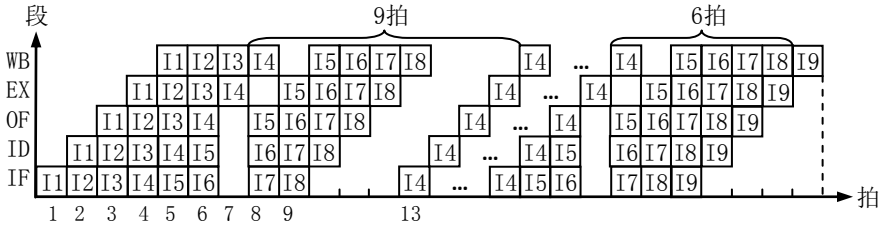


则程序执行时间为 $5\Delta t+(3-1)\Delta t+8\Delta t\times 100+\Delta t=808\Delta t$；

（2）控制相关处理采用静态分支预测法解决时，由于总是预测转移不发生，对 JNZ 指令预测错误导致流水线停顿 4 拍：

指令 \ 拍	1	2	3	4	5	6
I8：JNZ LOOP1	IF	ID	OF	EX	WB	
I9：???		IF	ID	OF		IF
			IF	ID		
				IF		

上述程序执行时流水线时-空图如下：



则程序执行时间为 $5\Delta t+(3-1)\Delta t+9\Delta t\times 99+6\Delta t+\Delta t=900\Delta t$。

第 6 章

1. 解释概念或术语：总线、系统总线、通信总线、总线宽度、总线带宽、总线周期、总线传输周期、地址期、数据期、主设备、从设备、总线仲裁、总线复用、总线桥。

答：略。

2. 为什么现代计算机中广泛采用总线结构实现部件互连？

答：计算机中功能部件的连接方式有分散连接和总线连接两种，前者的优点是多对部件可同时通信、通信性能好，缺点是可扩展性差；后者的优点是可扩展性好，缺点多对部件间只能分时通信。

随着计算机应用领域的不断扩大，I/O 设备的种类和数量也越来越多，可扩展性成为计算机系统必备的特性，而总线连接的缺点又可以通过相关技术得以缓解，因此现代计算机中广泛采用总线结构实现部件互连。

3. 总线有哪些特点？分别包含哪些内容？

答：（1）总线的特性有物理特性、功能特性、电气特性、时间特性 4 个方面。

（2）物理特性又称机械特性，指总线上部件在物理连接时的一些特性；功能特性指每根信号线在总县操作过程中的功能；电气特性指每根信号线上的信号方向及表示信号有效的电平范围；时间特性又称逻辑特性，指在总线操作过程中所有信号线上信号的时序关系约定。

4. 某 32 位宽度的同步总线中，总线时钟频率为 66MHz，若每个总线传输周期需要 4 个时钟周期，请计算该总线带宽。若想提高总线带宽，可采取哪些措施？

解：（1）由于每次总线传输都需要 4 个时钟周期，故总线传输次数/秒＝66MHz/4＝16.5MHz，总线带宽＝32bit×16.5MHz=528Mb/s=66MB/s；

（2）增加同步总线的数据宽度，提高总线时钟的频率，采用猝发传输模式等都可以提高总线带宽。

5. 某总线时钟频率为 33.3MHz，支持猝发传输模式，连续传输的速度是每个时钟一个数据，若总线的数据传输率为 532.8MB/s，寻址空间为 4G 个地址，问该总线的地址总线和数据总线宽度各为多少？

解：由于该总线支持猝发传输模式，故最大总线传输次数/秒＝33.3MHz/1＝33.3MHz；

设总线的数据总线宽度为 DN，则 532.8MB/s＝DN×33.3MHz，DN＝16B＝128bit；

地址总线宽度为 $\log_2(4G)=\log_2(4\times2^{30})=32bit$。

故该总线的地址总线和数据总线宽度各为 32 位和 128 位。

6. 一个总线周期包含哪些阶段？各阶段主要完成的工作是什么？

答：一个总线周期包含总线申请及仲裁、寻址、数据传送和结束 4 个阶段。

总线申请及仲裁阶段：总线仲裁机构根据主设备的总线请求信号，按照一定算法，决定将下个传输周期的总线使用权授予哪个申请者；

寻址阶段：获得总线使用权的主设备发出本次访问的目标设备地址和命令，从设备根据总线上的地址及命令，主动判别自身是否为目标设备，被选中时响应总线传输操作；

数据传送阶段：根据主设备发出的操作命令，主设备和从设备进行数据交换，数据由源设备发出，经总数据总线流入目标设备；

结束阶段：数据传输结束时，主设备、从设备均从总线上撤除自己所发出的信号，让出总线使用权，本次总线传输周期结束。

7. 为什么要进行总线仲裁？链式查询方式仲裁时，各主设备的仲裁逻辑是什么、有什么特点？计数器定时查询方式仲裁时，查询为什么要定时进行？如何实现循环优先级仲裁？

答：（1）由于总线是多个部件共用的传输介质，故同时只允许一个部件发送信息；为解决多个主设备同时竞争总线控制权的问题，必须要进行总线仲裁，用某种方式选择某个主设备为下个总县传输周期的总线使用权拥有者。

（2）链式查询方式仲裁时，各主设备的仲裁逻辑是 BSi＝BGi_{IN}&BRi、BGi_{OUT}＝BGi_{IN}&$\overline{BRi}$，即该设备被询问、且有总线请求时获得下个总线传输周期的总线使用权；

链式查询方式仲裁采用的是静态优先级策略，所需仲裁信号线最少，可扩展性强，但对电路故障很敏感，容易产生断链现象。

（3）计数器定时查询方式仲裁时，没有请求的主设备不会产生任何响应信息，因此必须询问必须定时切换，以保证轮询工作方式的有效进行；

每次仲裁时，若均从 0 开始计数，则实现的是静态优先级仲裁策略；若均从上次仲裁结束时的计数值继续开始计数，则实现的是循环优先级仲裁策略。

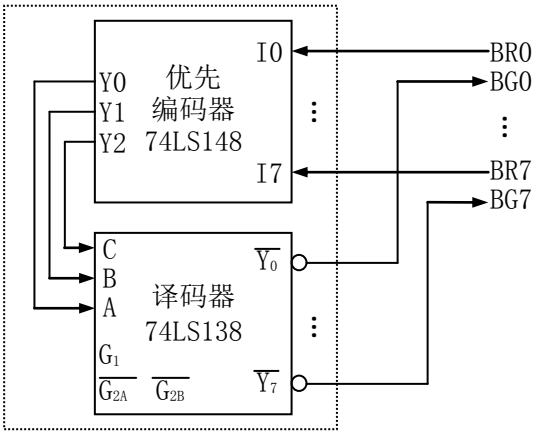
8. 图 6.4～图 6.6 中，总线仲裁机构是否可不连接控制总线？为什么？

答：链式查询方式、计数器定时查询方式的总线仲裁机构可以不连接控制总线，因为这两种方式通过信号线 BS 表示仲裁何时结束，连接控制总线只是可以实现隐藏式仲裁，但要求各信号功能及时序必须有所改变；

而独立请求方式的总线仲裁机构必须连接控制总线，因为这种方式没有 BS 信号线，只能通过总线状态得知何时开始与结束仲裁，由于仲裁时长固定，故常采用隐藏式仲裁方式。

9. 画出独立请求式仲裁时，BR0→BR7 降序的优先级形成电路图。

答：BR0→BR7 优先级降序可用优先编码器实现，输出为当前最高优先级对应请求；再用译码器输出仲裁结果即可：



10. 某 64 位总线支持猝发传输模式，每个时钟周期都能传输数据或地址，总线传输周期由 3 个时钟的地址期、若干个数据期组成。若存储器数据传输速度为每两个时钟周期才传输 64 位数据，每次访存最多传输 4 次数据。请计算下列两种应用情况下，总线和存储器所能提供的最大带宽各是多少？

（1）每次总线操作只传输 32 位数据。

（2）每次总线操作都传输包含 4 个数据期的数据块。

解：设总线始终频率为 f，

（1）对总线而言，每个总线传输周期由 3 个 CLK(地址期)及 1 个 CLK(数据期)组成，

总线最大带宽为 32bit×(f/4)＝8f (bps)，

对存储器而言，每个总线传输周期由 3 个 CLK(地址期)及 2 个 CLK(数据期)组成，

存储器的最大带宽为 32bit×(f/5)＝6.4f (bps)；

（2）对总线而言，每个总线传输周期由 3 个 CLK(地址期)及 4 个 CLK(数据期)组成，

总线最大带宽为(64bit×4)×(f/7)＝36.57f (bps)，

对存储器而言，每个总线传输周期由 3 个 CLK(地址期)及 8 个 CLK(数据期)组成，

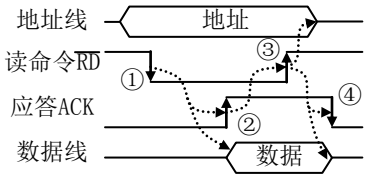
存储器的最大带宽为(64bit×4)×(f/11)＝23.28f (bps)。

11. 异步通信协议有哪几个阶段？简述异步通信时各阶段的具体操作。

答：（1）异步协议包括请求、响应、撤消请求、撤消响应 4 个阶段。

（2）请求阶段：主设备在有传输需求、从设备就绪时，发出操作请求信号 REQ；

响应阶段：从设备收到请求信号后进行相应操作，操作完成时发出应答信号 ACK；



撤消请求阶段：主设备收到应答信号后接收操作结果，然后撤消操作请求信号；

撤消响应阶段：从设备收到请求信号撤消后也撤消应答信号，一次异步通信结束。

12. 在起止式异步串行通信协议中，若信息格式为 1 个起始位、7 个数据位、1 个停止位，约定波特率为 3600bps，请问每秒最多能传输多少个字符？

解：由题意，传送一个字符需要 1＋7＋1＝9 位，

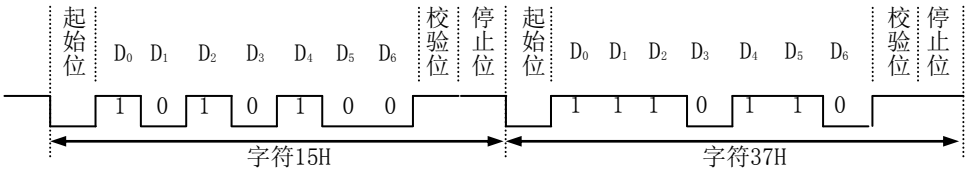
故每秒最多能传输的字符数为 3600/(1＋7＋1)＝400 个。

13. 若起止式异步串行通信协议的信息格式为 1 个起始位、7 个数据位、1 个偶检验位、1 个停止位，若连续发送字符 15H 及 37H，请画出数据信号线 D 上的传送波形图。

解：15H 的偶校验位为 0＋0＋1＋0＋1＋0＋1＝1，

37H 的偶校验位为 0＋1＋1＋0＋1＋1＋1＝1；

根据信息格式约定，每个字符需要 1＋7＋1＋1＝10 位，字符数据中低位先传送，D 上信号时序如下图所示：



14. 采用不互锁方式进行异步通信时，若并行传送方式的数据宽度为 16 位，传送的频率不固定，可以省略请求信号线吗？请说明理由。

答：不可以省略请求信号线。

由于异步不互锁方式通过口头约定实现操作时长估计、通过请求信号表示有无操作。理论上讲，请求信号与数据信号可分时用一根信号线上表示，但性能会成倍下降。当采用串行传送方式通信时，若多个数据位捆绑传送，请求信号与数据信号分时表示带来的性能损失很小，可以省略请求信号线；若单个数据位单独传送，请求信号与数据信号分时表示带来的成倍性能损失，导致不可以省略请求信号线；而采用并行传送方式通信时，通常每根数据线每次传送一个数据位，更不可能省略请求信号线了。

15. 为什么说半同步定时方式同时具备了同步定时和异步定时的优点？

答：半同步定时方式在同步定时方式的基础上，增设了一条等待（$\overline{\text{WAIT}}$）信号线。

当$\overline{\text{WAIT}}$信号无效时，主从设备以同步定时方式工作，所有信号在 CLK 的边沿发出，保持了同步方式双方配合简单、传输频率高的优点；当$\overline{\text{WAIT}}$信号有效时，主从设备以异

步定时方式工作，从设备完成操作时撤消 $\overline{\text{WAIT}}$，主设备在 CLK 边沿转入同步方式，保持了异步方式设备速度可相差较大的特点。

因此，半同步定时方式同时具备了同步定时和异步定时的优点。由于以同步定时方式为基础，故异步定时方式的传输距离可较长的特点无法体现出来。

16. PCI 总线的信号线中，PCI 总线采用的总线仲裁方式是什么？从设备比主设备少了哪两根信号线？PCI 总线有哪些特点？

答：PCI 总线具有集中式的总线仲裁机构，采用的是独立请求总线仲裁方式。

由于 PCI 总线采用独立请求仲裁方式，故从设备比主设备少了总线请求、总线允许两根信号线。

PCI 总线的特点较多，主要有独立于处理器、支持多主设备、功能较强(操作类型多/即插即用)、性能较高(猝发传输/半同步方式)等特点。

17. 简述单总线和多总线结构的特点。说明在总线桥中集成 I/O 接口等部件有什么好处？

答：（1）单总线结构的特点是控制简单、可扩展性强，但系统工作效率较低，总线易成为计算机系统的瓶颈。在多总线结构中，不同速度的设备连接在不同的总线上，各总线通过总线桥芯片进行连接，其特点是在不影响可扩展性的同时，可提高总线传输性能，如不同总线速度不同、多条总线可以同时传输。

（2）总线上部件必须按照总线标准进行传输，而在总线桥中集成 I/O 接口等部件，有三个好处：一是可减少总线长度；二是这些部件间传输可不受总线标准约束，提高传输性能；三是这些部件间传输可不使用总线，减少总线使用率。

第 7 章

1. 解释概念或术语：数据传输率、NRZ、RAID、I/O 接口、I/O 端口、中断、I/O 中断、中断请求、中断响应、中断服务、中断返回、向量中断、中断向量表、中断判优、中断嵌套、中断屏蔽、DMA、周期窃取、DMA 预处理、DMA 后处理、通道。

答：略。

2. 现代计算机中，为什么 I/O 设备通常通过总线与主机连接？

答：随着计算机应用的普及，现代计算机中 I/O 设备种类越来越多、速度大不相同，并且要求计算机可随时可接入这些设备。由于总线连接方式具有可扩展性好、能够实现操作标准化等优点，可以满足相关应用需求，故 I/O 设备通常通过总线与主机连接。

3. 简述 I/O 设备有哪两种编址方式？它们对指令系统及总线信号线有哪些影响？

答：I/O 设备有统一编址和独立编址两种编址方式。

统一编址方式时存储器和 I/O 设备地址不重叠，只通过地址即可区分这两种部件，故指令系统无需任何变化，总线信号也不受任何影响(只需 MEMR#、MEMW#两根控制线)，只是系统可扩展性不够好、指令格式较长；

独立编址方式的存储器和 I/O 设备地址重叠，只通过地址无法区分这两种部件，故指令系统需增设两条 I/O 指令，总线信号相应地也需增设 IOR#、IOW#两根控制线。

4. I/O 设备与主机交换信息时，共有哪几种控制方式？简述其特点。

答：I/O 设备与主机交换信息时，共有程序查询、程序中断、DMA、通道等 4 种方式。

程序查询方式中，CPU 不停地查询 I/O 设备状态，只有在设备就绪时才进行信息传送，其特点是 I/O 设备及主机组成简单，但 CPU 工作效率较低(CPU 与外设串行工作)；

程序中断方式中，CPU 启动 I/O 设备后，继续执行现行程序，I/O 设备就绪后提出请求时，才响应请求进行信息传送，其特点是 CPU 工作效率较高（CPU 与外设部分并行工作），但 I/O 设备及 CPU 需增设与中断相关软硬件；

DMA 方式中，I/O 设备直接与存储器进行信息传送，传送无需 CPU 干预、只需 CPU 让出总线使用权，CPU 仅负责传送准备及结束处理工作，其特点是 CPU 工作效率在程序中断方式基础上有进一步提高，但 I/O 设备硬件组成更复杂、CPU 需增设 DMA 请求/响应机制。

通道方式中，CPU 仅负责将传送需求编制成通道程序，其余工作基本全部由通道独立完成，其特点是 CPU 工作效率在 DMA 方式基础上有进一步提高，但需增设通道处理器。

从上述传送控制方式可见，系统性能提高是以增加硬件成本为代价的，但性能/价格有明显上升。

5. 对某 I/O 设备，简述其驱动器、控制器及适配器的功能与组成。

答：I/O 设备通常由设备部件及设备控制器组成，设备部件由机、电、光、磁等器件组成，以实现设备的约定功能。

设备驱动器是一种设备部件，对某些需多个设备部件组合来实现功能的 I/O 设备，其中的一些设备部件常称为设备驱动器，如磁盘存储器的设备部件由磁盘和磁盘驱动器组成，驱动器负责实现电信号与机械、光、磁等信号的转换，由机械、光、磁、电等器件组成。

设备控制器负责外部的设备操作命令(数字信号)与内部的设备部件控制信号或驱动信号的转换，设备控制器由各种信号及时序转换电路组成。

设备适配器又称为 I/O 接口，主要实现标准化的总线操作信号与非标准化的 I/O 设备操

作命令间的转换，由设备选择电路、锁存器或寄存器、控制逻辑电路、信号转换逻辑电路等组成。

6. 某图形显示器最大分辨率为 1280×1024，最大灰度级为 32 位时，为充分发挥显示器性能，显示卡中 VRAM 的容量应为多少？

解：欲充分发挥显示器性能，显示卡中 VRAM 至少应能存储最大分辨率的各像素点信息。

因此，显示卡中 VRAM 容量至少应为：1280×1024×32bit＝5120KB。

7. 图 7.19 中打印机与打印机接口所采用的是哪种联络方式？设置了响应信号后再设置忙信号的好处是什么？

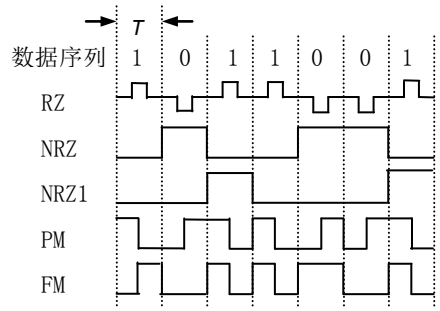
答：采用的是异步联络方式，通过“选通”、“响应”信号实现异步操作的握手时序。

异步联络方式中，对设备的各种操作均是串行的，为提高设备性能，某些操作是可以重叠进行的，如打印当前字符与接收下个字符数据可以重叠。而当前时刻重叠操作能否进行，需要通过状态信号表示。

打印机中设置“忙”信号，可以使字符打印与字符数据接收重叠进行，隐藏了字符数据接收的延迟，提高了打印机工作效率。

8. 画出 RZ、NRZ、NRZ1、PM、FM 写入数字串 1011001 的写电流波形图。

解：几种编码方式的写电流波形图如下图所示，NRZ 及 NRZ1 编码方式写第一个“1”时的写电流与前一位有关，暂定为如此：



9. 某磁盘组有 6 个盘片，最外两侧盘面为保护面、不记录信息。盘片存储区域内径为 22cm，外径为 33cm，磁道间距最小为 0.25cm，磁道位密度为 1600b/cm。

（1）此磁盘组的存储容量是多少？

（2）当磁盘转速为 5400r/min，且同时仅一个磁头工作时，数据传输率是多少？

解：（1）每个盘面磁道数＝(33-22)÷2÷0.25=22 个，

每个磁道信息量＝2×3.14×11×1600＝110528 bit＝13816 B，

磁盘组存储容量＝(6×2－2)×22×13816＝3039520 B；

（2）数据传输率＝记录密度×盘面转速＝13816×(5400÷60)＝1243440 B/s。

10. 简述 I/O 接口的基本功能及接口硬件的基本组成，分析两者间关系。

答：I/O 接口的基本功能主要有设备寻址功能、数据缓冲功能、操作中转功能、信号转换功能及设备状态监视功能。

I/O 接口硬件主要由设备选择电路、寄存器、控制逻辑电路、信号转换逻辑电路等组成。

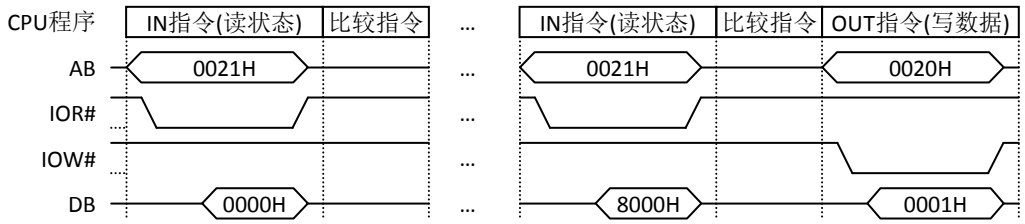
I/O 接口硬件中，设备选择电路主要实现设备寻址功能；寄存器主要实现数据、所接收操作、所监视设备状态的暂存功能；控制逻辑电路主要实现操作中转功能，同时负责其它功能的时序控制；信号转换逻辑电路主要实现信号转换功能。

由此可见，I/O 接口的各项功能由 I/O 接口硬件中相应组成部分实现，其中控制逻辑电路是核心，负责各组成部分的控制与协调。

11. 若 I/O 接口连接在系统总线上，对 I/O 接口操作的控制总线信号为 IOR#和 IOW#，某 I/O 接口数据口、状态口的 I/O 端口地址分别是 0020H、0021H，假设指令执行过程的访存操作均不产生使用系统总线(即均在 Cache 中命中)，请画出采用程序查询方式向该 I/O 接口写数据 0001H 时，总线(AB、CB、DB)上的信号及时序。

解：程序查询方式中，CPU 不停地查询设备状态、测试并比较设备状态，直到设备就绪时才向 I/O 接口写数据。

读设备状态指令所用 I/O 端口地址为 0021H，比较指令不使用总线、总线空闲(IOR#、IOW#均无效)，写数据指令所用 I/O 端口地址为 0020H。假设状态口最高位为 1 时表示设备就绪，则程序查询方式向该 I/O 接口写数据 0001H 时总线上的信号及时序如下图：



12. 在程序(指令串)执行过程中，I/O 中断请求什么时候可能产生？为什么说 I/O 中断请求是可屏蔽中断请求？如何表示当前是否处于屏蔽 I/O 中断请求状态？何时响应 I/O 中断请求，硬件实现成本最低？

答：I/O 设备请求数据传送的中断称为 I/O 中断。

I/O 设备准备就绪即可产生中断，相对于 CPU 的程序执行，I/O 中断请求在任何时候均可能产生，因为两个部件独立工作。

可屏蔽中断请求只可稍后处理的中断请求，由于 I/O 中断请求稍后处理不会影响正确性，如键被按下、移动鼠标等，因此 I/O 请求属于可屏蔽中断请求。

为表示当前是否处于屏蔽 I/O 中断请求状态，通常在 CPU 中设置一个称为“中断允许”的标志位（记为 IF），用 IF＝0 和 IF＝1 分别表示当前是屏蔽还是允许 I/O 中断请求。

由于 CPU 的工作流程是循环地取出并执行指令，而指令执行过程内部通常前后关联、不易分割，因此，I/O 中断请求(可屏蔽请求)放在两轮循环之间，即两条指令执行过程之间处理时硬件实现成本最低。

13. 对于向量中断，简述中断响应的功能及各个阶段的具体任务。并说明 I/O 中断请求被响应的条件。

答：中断响应指 CPU 从当前程序转入中断服务程序的过程，主要包括识别中断源、保存现场、获得中断服务程序入口地址、转入中断服务阶段 4 项工作。

识别中断源的任务是从当前所有中断请求中选择一个最紧急的中断请求，并取得所选中断请求对应中断源的中断向量地址；保存现场的任务是保存程序返回点硬件现场，通过关中断(即 IF←0)可实现默认的单重中断方式；获得中断服务程序入口的任务是用所选中断源的中断向量地址查中断向量表 IVT，得到所选中断请求对应的中断服务程序入口地址（中断向量)；转入中断服务阶段的任务最简单，只需将所获得的中断服务程序入口地址置入 PC 即可。CPU 按(PC)执行指令即进入了中断处理阶段。

I/O 中断请求被响应的条件：中断请求信号有效、无 DMA 请求或更紧急的中断请求、“中断允许”位 IF＝1、当前指令结束(End 信号有效)时。

14. 为什么中断返回用机器指令表示？通常中断返回点如何获得？若不采用上述方法，有哪些麻烦？

答：由于中断服务的具体功能通过中断服务程序实现，而中断服务程序何时结束只能在

程序中通过中断返回指令表明，因此中断返回用机器指令表示。

现代计算机系统中，为减小中断的实现复杂度，中断返回后总是执行一条新指令，即中断返回点是中断返回处的指令地址。由于中断请求类型决定了中断返回时的指令地址(当前指令/下条指令地址)，因此，中断响应时保存硬件现场通常保存中断返回点、而非中断断点。

若中断响应时保存的不是中断返回点，则中断返回时需根据断点及中断源类型来计算中断返回点，一是需额外保存中断源类型、增加了硬件成本，二是返回时再计算、降低了返回性能。

15. 某系统采用中断方式对设备进行数据采样，中断处理时读取数据，并存至主存缓冲区中，该中断处理需要 2ms。此外，缓冲区中每存储 1000 个数据，主程序就要将其取出并处理，这个处理时间需 50ms。试问该系统每秒可以采样多少个数据？

解：采样并处理 1000 个数据总时间为：1000×2＋50＝2050ms＝2.05s，

系统每秒采样数据个数＝1000/2.05＝487.8 个/秒。

16. 某计算机的主频为 50MHz，与某个数据宽度为 32 位、数据传输率为 32Kbps 的设备进行数据 I/O 时，程序查询方式的一次查询操作需 200 个时钟周期，一次数据传送操作需 100 个时钟周期，程序中断方式的一次中断处理需 500 个时钟周期。

（1）若程序查询方式中，CPU 每查询 10 次就可以传送一次数据，此时 I/O 占 CPU 时间的百分比是多少？

（2）程序中断方式中，I/O 占 CPU 时间的百分比又是多少？

解：该设备每秒最多可进行数据 I/O 的次数为 32Kbps/32bit＝1000 次/秒，

（1）每次 I/O 所占 CPU 时间为(200×10＋100)/(50×10^6)＝42×10^{-6}s，

程序查询方式中，I/O 占 CPU 时间的百分比为[1000×(42×10^{-6})]/1＝4.2%；

（2）每次 I/O 所占 CPU 时间为 500/(50×10^6)＝10×10^{-6}s，

程序中断方式中，I/O 占 CPU 时间的百分比为[1000×(10×10^{-6})]/1＝1.0%。

17. 在数据传送过程中，I/O 接口应何时产生中断请求？如何使接口既支持中断方式、又支持程序查询方式工作？如何实现中断响应时撤销中断请求？

答:若 I/O 接口用状态口中 RD 位表示 I/O 设备状态,用触发器 INTR 表示中断请求状态，

（1）I/O 接口应在 I/O 设备准备就绪(或传送完成)时提出中断请求，

即 RD←1 时、INTR←1，即 RD 从 0→1 时、INTR 从 0→1；

（2）应在 I/O 接口的控制口中增设“允许中断”位(EI)，EI＝1 时表示当前采用的是中断方式，EI＝0 时表示当前采用的是程序查询方式；

（3）I/O 接口的中断响应(记为$\overline{\text{INTA}}$)信号有效时，撤消中断请求(CPU 准备处理该 I/O 接口的中断请求)，即$\overline{\text{INTA}}$ ←0(即 1→0)时，INTR←0(即 1→0)，由于此时 RD＝1，故 INTR 用触发器实现，产生中断请求时 S 引脚 0→1，撤消中断请求时 R 引脚 0→1。

18. 相对于串行判优，并行判优有哪些优缺点？为了扬长避短，实际应用中的 I/O 中断请求的常用连接及判优方法是什么？

答：串行判优的优点是可扩展性好，缺点是响应速度慢、优先级不可改变、存在断链问题、中断向量地址需各 I/O 接口预先约定好；并行判优不需要 I/O 接口参与，优点是响应速度快、优先级可变、不存在断链现象、中断向量地址统一形成等，缺点是可扩展性不好。

为解决可扩展性问题，通常采用串行判优与并行判优相结合方式，即 I/O 请求采用独立请求方式连接到中断控制器，实现并行判优；中断控制器采用共享连接方式连接到 CPU，实现串行判优；通常中断控制器还支持级联方式进一步提高可扩展性。

19. 什么是多重中断？如何表示当前是否是多重中断？实现多重中断的基本条件有哪

些？

答：多重中断指中断过程中可以响应新的 I/O 中断请求，暂停执行现行中断程序、转去执行新中断程序的中断方式，又称中断嵌套。

由于 I/O 中断请求属可屏蔽中断请求，通常用 IF 位表示当前是否处于屏蔽 I/O 中断请求状态，不屏蔽时才可能实现多重中断方式效果，故通常用 IF 表示当前是单重/多重中断。

实现多重中断的基本条件有：中断判优逻辑中须设置正在服务请求、尚未服务请求的请求寄存器、排队器(编码器)、比较器，以及实现 I/O 指令复位某中断请求功能的硬件；CPU 的中断响应机构中需用后援寄存器堆栈代替后援寄存器保存多的中断现场。

20. DMA 方式有何特点？由谁实现传送过程控制？需要哪些条件？

答：DMA 方式的主要特点有：传送过程不占用 CPU 时间，可按总线周期速度传送，采用批量传送方式。

传送过程由 DMA 接口控制，DMA 接口是一种功能更强的 I/O 接口。

DMA 方式需两方面的支持：一是需在主存中开辟好缓冲区，以支持批量传送；二是传送时需 CPU 让出存储器总线控制权，现代计算机的层次结构存储系统提供了较好的支持。

21. 当 DMA 接口分别采用暂停 CPU 访问方式与周期挪用方式传送时，请说明 DMA 接口在申请总线控制权方面有何不同。

答：DMA 接口采用暂停 CPU 访问方式工作时，开始传送时，DMA 接口向 CPU 提出总线使用请求，DMA 接口获得总线使用权后，在所有数据全部传送结束前，均不放弃总线使用权(CPU 无法访问总线)，故称为暂停 CPU 访问方式。

DMA 接口采用周期挪用方式工作时，每当 I/O 设备准备就绪(准备传送单个数据)时，DMA 接口才向 CPU 提出总线使用请求，DMA 接口获得总线使用权后控制传送，然后立即向 CPU 交回总线使用权，I/O 设备准备就绪时再申请，故称为(总线)周期挪用方式。

在申请总线控制权方面的主要不同点在于一次申请还是多次申请，周期挪用方式对 CPU 的影响较小，CPU 效率得到了较好的发挥。

22. 请说明 DMA 接口为了实现其控制的批量传送，硬件电路中应具有哪些部件？

答：DMA 接口电路中应具有主存地址计数器(MAC)、传送字数计数器(WC)，以支持一个字/次的批量数据传送；应具有传送完毕(WC←0)的中断请求逻辑，请求 CPU 进行后续处理。

23. 结合 DMA 接口电路，简述 DMA 方式传送过程中各个阶段的具体工作。

答：DMA 方式传送过程包括预处理、数据传送及后处理三个阶段。

预处理阶段，CPU 向 DMA 接口写入本次的传输参数，并启动 I/O 设备，然后继续执行现行程序。传输参数包括主存缓冲区首址、传送字数、传送方向及传送方式等参数(与 I/O 设备无关)，启动 I/O 设备指设置操作参数(与 I/O 设备有关)。

数据传送阶段，DMA 接口按照传送方式规则、使用循环方式、组织与控制数据传送(一个字/次)。以周期窃取方式的 DMA 读为例：①I/O 设备就绪时，数据以送至数据缓冲寄存器(BR)中，DMA 接口向 CPU 提出总线使用请求(HRQ)；②CPU 在当前访存周期结束时，将总线使用权授予给 DMA 接口(HLDA)；③DMA 接口通过总线实现主存-I/O 设备的数据传送操作，即 AB←(MAC)、CB←MEMW、DB←(BR)；④DMA 接口实现循环，即 MAC←(MAC)＋1、WC←(WC)－1，若 WC≠0，撤销 HRQ(交回总线使用权)、等待 I/O 设备就绪(即转①)，若 WC＝0，产生 EOP(计数器溢出)、向 CPU 提出中断请求(传送已结束)。

后处理阶段，CPU 响应 DMA 接口的中断请求，完成传送结束处理工作，如数据校验、启动下次 DMA 传送等。

24. 比较程序中断方式与 DMA 方式的区别。

答：DMA 方式与程序中断方式主要有如下区别：

①DMA 方式的传送控制由 DMA 接口用硬件实现、CPU 每干预一次传送一批数据，中断方式由 CPU 用程序实现、CPU 每干预一次传送一个字；

②DMA 方式的数据传送需 CPU 暂停访问主存、无其它开销，中断方式需 CPU 暂停现行程序、有中断响应开销；

③DMA 方式的最大传送速度为一个字/存储周期，中断方式为一个字/指令周期；

④DMA 方式的 DMA 请求优先级高于中断方式的中断请求，CPU 在当前存取周期结束时响应 DMA 请求，CPU 在一条指令完成时响应中断请求；

⑤DMA 方式无传送异常处理能力，中断方式可以处理传送异常(中断程序实现)。

25. 某 CPU 的主频为 500MHz、CPI 等于 5，假设某外设数据传输率为 0.5MB/s，采用中断方式进行数据传送，以 4B 长度为传送单位，对应的中断程序共 18 条指令，中断处理的其它开销相当于 2 条指令时间。

（1）中断方式下，CPU 用于该外设 I/O 的时间占整个 CPU 时间的百分比？

（2）若外设数据传输率提高到 5MB/s，改用 DMA 方式传送，每次 DMA 传送的块大小为 5000B，DMA 预处理及后处理共需 500 个时钟周期，假设 CPU 与 DMA 接口间无访存冲突，则 CPU 用于该外设 I/O 的时间百分比？

解：（1）CPU 每次中断开销＝(18＋2)×5T_C＝100T_C＝100/(500×10^6) ＝0.2×10^{-6}s，

外设的中断请求频率＝0.5MB/s÷4B＝125000Hz，中断请求间隔＝1/125000＝8×10^{-6}s，

I/O 时间占 CPU 时间百分比＝(0.2×10^{-6})÷(8×10^{-6})＝2.5%。

（2）CPU 每次 DMA 处理开销＝500T_C＝500/(500×10^6) ＝1×10^{-6}s，

外设的 DMA 请求频率＝5MB/s÷5000B＝1000Hz，DMA 请求间隔＝1/1000＝1×10^{-3}s，

I/O 时间占 CPU 时间百分比＝(1×10^{-6})÷(1×10^{-3})＝0.1%。